Intermediate System Administration for the Solaris™ 9 Operating Environment SA-239

Student Guide

Sun Microsystems, Inc.
UBRM05-104
500 Eldorado Blvd.
Broomfield, CO 80021
U.S.A.

Revision A.2

Export Commodity Classification Number (ECCN) assigned: 12 December 2001

SA239_A2_0903_2_F

Please
Recycle

Adobe PostScript

Table of Contents

About This Course

Instructional Goals

Upon completion of this course, you should be able to:

- Manage file systems
- Install software
- Perform system boot procedures
- Perform user and security administration
- Manage network printers and system processes
- Perform system backups and restores

Course Map

The course map enables you to see what you have accomplished and where you are going in reference to the instructional goals.

Managing File Systems

- Introducing the Solaris™ OE Directory Hierarchy
- Managing Local Disk Devices
- Managing the Solaris OE File System
- Performing Mounts and Unmounts

Installing Software

- Installing the Solaris™ 9 OE
- Performing Solaris 9 OE Package Administration
- Managing Software Patches on the Solaris 9 OE

Performing System Boot Procedures

- Executing Boot PROM Commands
- Performing Boot and Shutdown Procedures

Performing User and Security Administration

- Performing User Administration
- Performing System Security

Managing Network Printers and System Processes

- Configuring Printer Services
- Using Print Commands
- Controlling System Processes

Performing System Backups and Restores

- Performing File System Backups
- Performing File System Restores
- Backing Up a Mounted File System With a UFS Snapshot

Topics Not Covered

This course does not cover the following topics. Many of these topics are covered in other courses offered by Sun Educational Services:

- Basic UNIX® commands – Covered in SA-119: *UNIX® Essentials Featuring the Solaris™ 9 Operating Environment*
- The `vi` editor – Covered in SA-119: *UNIX® Essentials Featuring the Solaris™ 9 Operating Environment*
- Basic UNIX file security – Covered in SA-119: *UNIX® Essentials Featuring the Solaris™ 9 Operating Environment*
- JumpStart™ procedure– Covered in SA-299: *Advanced System Administration for the Solaris™ 9 Operating Environment*
- Network File System (NFS) environment configuration – Covered in SA-299: *Advanced System Administration for the Solaris™ 9 Operating Environment*
- Naming services – Covered in SA-299: *Advanced System Administration for the Solaris™ 9 Operating Environment*
- Troubleshooting – Covered in ST-350: *Sun™ Systems Fault Analysis Workshop*
- System tuning – Covered in SA-400: *Solaris™ System Performance Management*

Refer to the Sun Educational Services catalog for specific information and registration.

How Prepared Are You?

To be sure you are prepared to take this course, can you answer yes to the following questions?

- Perform basic UNIX tasks
- Understand basic UNIX commands
- Use the vi text editor
- Interact with a windowing system

Introductions

Now that you have been introduced to the course, introduce yourself to the other students and the instructor, addressing the items shown below:

- Name
- Company affiliation
- Title, function, and job responsibility
- Experience related to topics presented in this course
- Reasons for enrolling in this course
- Expectations for this course

How to Use Course Materials

To enable you to succeed in this course, these course materials employ a learning module that is composed of the following components:

- Objectives – You should be able to accomplish the objectives after completing a portion of instructional content. Objectives support goals and can support other higher-level objectives.
- Lecture – The instructor will present information specific to the objective of the module. This information will help you learn the knowledge and skills necessary to succeed with the activities.
- Activities – The activities take on various forms, such as an exercise, self-check, discussion, and demonstration. Activities are used to facilitate mastery of an objective.
- Visual aids – The instructor might use several visual aids to convey a concept, such as a process, in a visual form. Visual aids commonly contain graphics, animation, and video.

Note – Many system administration tasks for the Solaris™ Operating Environment can be accomplished in more than one way. The methods presented in the courseware reflect recommended practices used by Sun Educational Services.

Conventions

The following conventions are used in this course to represent various training elements and alternative learning resources.

Icons

Discussion – Indicates a small-group or class discussion on the current topic is recommended at this time.

Demonstration – Indicates a demonstration of the current topic is recommended at this time.

Note – Indicates additional information that can help students but is not crucial to their understanding of the concept being described. Students should be able to understand the concept or complete the task without this information. Examples of notational information include keyword shortcuts and minor system adjustments.

Caution – Indicates that there is a risk of personal injury from a nonelectrical hazard, or risk of irreversible damage to data, software, or the operating system. A caution indicates that the possibility of a hazard (as opposed to certainty) might happen, depending on the action of the user.

Typographical Conventions

`Courier` is used for the names of commands, files, directories, user names, host names, programming code, and on-screen computer output; for example:

> Use the `ls -al` command to list all files.
> `host1# cd /home`

`Courier bold` is used for characters and numbers that you type; for example:

> To list the files in this directory, type the following:
> `#` **`ls`**

`Courier italics` is used for variables and command-line placeholders that are replaced with a real name or value; for example:

> To delete a file, use the `rm` *`filename`* command.

`Courier italic bold` is used to represent variables whose values are to be entered by the student as part of an activity; for example:

> Type **`chmod a+rwx`** ***`filename`*** to grant read, write, and execute rights for *`filename`*.

Palatino italics is used for book titles, new words or terms, or words that you want to emphasize; for example:

> Read Chapter 6 in the *User's Guide*.
> These are called *class* options.

Module 1

Introducing the Solaris™ OE Directory Hierarchy

Objectives

Upon completing this module, you should be able to:

- Describe / (root) subdirectories
- Describe file components
- Describe file types
- Use hard links

The following course map shows how this module fits into the current instructional goal.

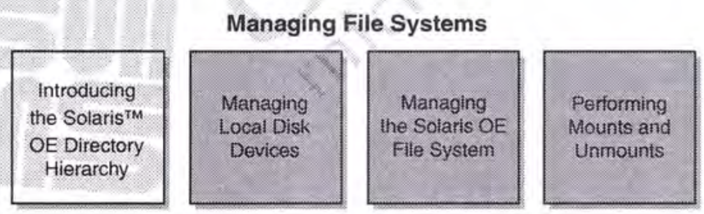

Figure 1-1 Course Map

Introducing / (root) Subdirectories

The directory hierarchy of the Solaris™ Operating Environment (Solaris OE) is organized for administrative convenience. Branches within this directory tree segregate directories that are used for different purposes. For example, directories exist to hold files that are private to the local system, files to share with other systems, and `home` directories.

Logically, all directories fall below the / (root) directory. Physically, however, directories can be located on a single file system or divided among multiple file systems. Every Solaris OE must have a `root` file system but can also have other file systems attached at points within the directory hierarchy. Most file systems are structures created on disk slices that contain or hold files and directories.

Note – Refer to `man -s5 filesystem` for information on file system organization.

Introducing Important System Directories

The Solaris OE consists of a hierarchy of critical system directories and files that are necessary for the operating system to function properly. The following is a list of some of the critical system directories and subdirectories that are found in the Solaris OE.

/	The root of the overall file system namespace.
/bin	A symbolic link to the /usr/bin directory. It is the directory location for the binary files of standard system commands.
/dev	The primary directory for logical device names. The contents of this directory are symbolic links that point to device files in the /devices directory.
/devices	The primary directory for physical device names.
/etc	The directory that holds host-specific configuration files and databases for system administration.
/export	The default directory for commonly shared file systems, such as users' home directories, application software, or other shared file systems.
/home	The default directory or mount point for a user's home directory.
/kernel	The directory of platform-independent loadable kernel modules that are required as part of the boot process.
/mnt	A convenient, temporary mount point for file systems.
/opt	The default directory or mount point for add-on application packages.
/platform	The directory of platform-dependent loadable kernel modules.
/sbin	The single-user bin directory that contains essential executables that are used during the booting process and in manual system-failure recovery.
/tmp	The directory for temporary files. This directory is cleared during the boot sequence.
/usr	The directory that contains programs, scripts, and libraries that are used by all system users. The directory name is an acronym for UNIX® system resources.
/var	The directory for varying files, which usually includes temporary, logging, or status files.

The following tables list primary subdirectories under key directories.

Table 1-1 Primary Subdirectories Under the /dev Directory

Subdirectory	Description
/dev/cua	Dial-out device files for UNIX-to-UNIX Copy Protocol (UUCP) and Point-to-Point Protocol (PPP)
/dev/dsk	Block disk devices
/dev/fbs	Frame buffer device files
/dev/fd	File descriptors
/dev/md	Logical volume management metadisk devices
/dev/pts	Pseudo terminal devices
/dev/rdsk	Raw disk devices
/dev/rmt	Raw magnetic tape devices
/dev/sound	Audio device and audio-device control files
/dev/term	Serial devices

Table 1-2 Primary Subdirectories Under the /etc Directory

Subdirectory	Description
/etc/acct	Configuration information for the accounting system
/etc/cron.d	Configuration information for the cron utility
/etc/default	Default information for various programs
/etc/inet	Configuration files for network services
/etc/init.d	Scripts for starting and stopping services, used while changing between run levels
/etc/lib	Dynamic linking libraries needed when the /usr file system is not available
/etc/lp	Configuration information for the printer subsystem
/etc/mail	Configuration information for the mail subsystem

Table 1-2 Primary Subdirectories Under the /etc Directory (Continued)

Subdirectory	Description
/etc/nfs	Configuration file for NFS server logging
/etc/opt	Configuration information for optional packages
/etc/rc#.d	Scripts that are when entering or leaving a specific run level
/etc/skel	Default shell initialization files for new user accounts

Table 1-3 Contents of the /usr Directory

Subdirectory	Description
/usr/bin	Standard system commands
/usr/ccs	C-compilation programs and libraries
/usr/demo	Demonstration programs and data
/usr/dt	Directory or mount point for Common Desktop Environment (CDE) software
/usr/include	Header files (for C programs, and so on)
/usr/java	Directories that contain Java™ technology programs and libraries
/usr/kernel	Platform-independent loadable kernel modules that are not generally required during the boot process
/usr/lib	Architecture-dependent databases, various program libraries, and binaries that are not invoked directly by the user
/usr/opt	Configuration information for optional packages
/usr/sbin	System administration commands
/usr/spool	Symbolic link to the /var/spool directory

Introducing File Components

All files in the Solaris OE make use of a file name and a record called an inode. Most files also make use of data blocks. In general, a file name is associated with an inode, and an inode provides access to data blocks.

Figure 1-2 shows the relationship between the file components.

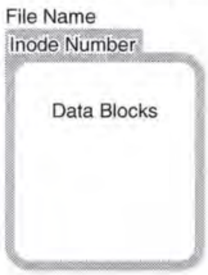

Figure 1-2 File Names, Inodes, and Data Blocks

File Names

File names are the objects most often used to access and manipulate files. A file must have a name that is associated with an inode.

Inodes

Inodes are the objects the Solaris OE uses to record information about a file. In general, inodes contain two parts. First, inodes contain information about the file, including its owner, its permissions, and its size. Second, inodes contain pointers to data blocks associated with the file.

Inodes are numbered, and each file system contains its own list of inodes. When a new file system is created, a complete list of new inodes is also created in that file system.

Data Blocks

Data blocks are units of disk space that are used to store data. Regular files, directories, and symbolic links make use of data blocks. Device files do not hold data.

Identifying File Types

The Solaris OE supports a standard set of file types that are found in nearly all UNIX-based operating systems. In general, files provide a means of storing data, activating devices, or allowing inter-process communication. Of the different types of files that exist in the Solaris OE, there are four main file types:

- Regular or ordinary files
- Directories
- Symbolic links
- Device files

Regular files, directories, and symbolic links all store one or more types of data. Device files do not store data. Instead, device files provide access to devices.

Use the `ls` command to distinguish different file types from one another. The character in the first column of information that the `ls -l` command displays indicates the file type.

The following examples, taken from a SPARC® technology Ultra™ 5 workstation, show partial listings of directories that contain a variety of different file types:

```
# cd /etc
# ls -l
total 428
drwxr-xr-x   2 adm      adm          512 Apr  3 10:42 acct
lrwxrwxrwx   1 root     root          14 Apr  3 11:05 aliases ->
./mail/aliases
drwxr-xr-x   2 root     bin          512 Apr  3 10:45 apache
-rw-r--r--   1 root     bin           50 Apr  3 10:45 auto_home
-rw-r--r--   1 root     bin          113 Apr  3 10:45 auto_master
(output truncated)
```

```
# cd /devices/pci@1f,0/pci@1,1/ide@3
# ls -l
total 0
brw-------   1 root     sys      136,  0 Apr  3 11:11 dad@0,0:a
crw-------   1 root     sys      136,  0 Apr  3 11:11 dad@0,0:a,raw
brw-------   1 root     sys      136,  1 Apr  4 11:06 dad@0,0:b
crw-------   1 root     sys      136,  1 Apr  3 11:11 dad@0,0:b,raw
(output truncated)
```

The character in the first column identifies each file type, as follows:

`-`	Regular files
`d`	Directories
`l`	Symbolic links
`b`	Block-special device files
`c`	Character-special device files

Regular Files

Perhaps the most common file types found in the Solaris OE are regular files, which enable the user to store many different types of data. Regular files can hold American Standard Code for Information Interchange (ASCII) text or binary data, including image data, database data, application-related data, and more.

There are many ways to create regular files. For example, a user could use the `vi` editor to create an ASCII text file, or a user could use a compiler to create a file that contains binary data. As another example, a user could use the `touch` command with a nonexistent file name to create a new, empty regular file.

Figure 1-3 shows a regular file called `file1`. As illustrated, the name `file1` is associated with inode number 1282. The data blocks associated with `file1` can hold one of many types of data, and the file could have been created in one of many different ways.

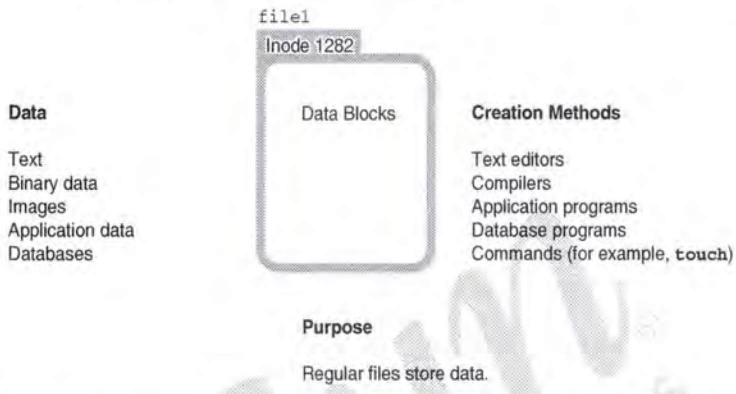

Figure 1-3 Regular Files

Directories

Directories store information that associates file names with inode numbers. Unlike regular files, which can hold many different types of data, directories only hold file name-to-inode associations.

A directory contains entries for files of all types that are logically found within that directory.

Figure 1-4 shows information about a directory called `dir1`. As illustrated in the figure, the name `dir1` is associated with inode number 4221. The data blocks associated with the `dir1` directory hold a list of file names and their associated inode numbers. The `mkdir` command is one way to create new directories.

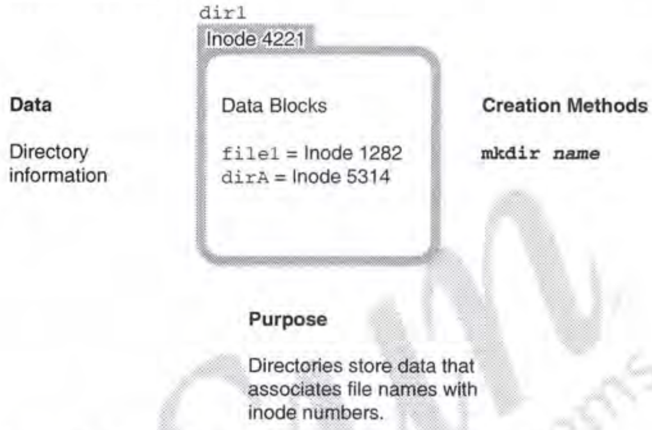

Figure 1-4 Directories

Think of the information that directories hold as a list. Each entry in this list accounts for one file name. If the file called `file1` and the directory called `dirA` were logically located in the directory called `dir1`, then the `dir1` directory would contain an entry that associates the name `file1` with inode number 1282 and an entry that associates the name `dirA` with inode number 5314.

Symbolic Links

A symbolic link is a file that points to another file. Like directories, which contain only directory information, symbolic links contain only one type of data.

A symbolic link contains the path name of the file to which it points. Because symbolic links use path names to point to other files, they can point to files in other file systems.

The size of a symbolic link always matches the number of characters in the path name it contains.

In the following example, the symbolic link called /bin points to the directory ./usr/bin. The size of the symbolic link is 9 bytes because the path name ./usr/bin contains nine characters.

```
# cd /
# ls -l bin
total 135
lrwxrwxrwx   1 root     root           9 Mar 22 10:39 bin -> ./usr/bin
```

Symbolic links can point to regular files, directories, other symbolic links, and device files. They can use absolute or relative path names.

The ln command with the -s option creates a symbolic link.

```
# ln -s file1 link1
```

Symbolic links direct read and write operations to the file to which they point. The preceding command shows how using link1 as a command-line argument causes the ln command to refer to the file called file1.

Figure 1-5 shows a symbolic link file called link1. As shown in the following figure, the link1 file is associated with inode number 3561. The data block for the link1 file contains the path name (./file1) to file1.

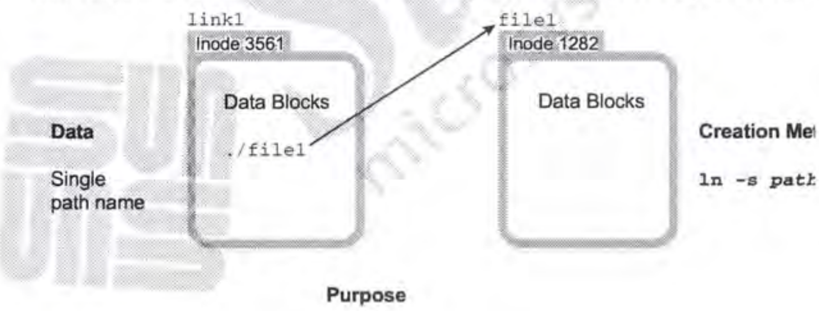

Figure 1-5 Symbolic Links

Device Files

A device file provides access to a device. Unlike regular files, directories, and symbolic links, device files do not use data blocks. Instead, the inode information of device files holds numbers that refer to devices. Use the `ls -l` command to display these numbers.

For example, a long listing of a regular file shows the file's size.

```
# cd /etc
# ls -al |more
total 510
drwxr-xr-x  51 root     sys          3584 Mar 14 07:35 .
drwxr-xr-x  29 root     root         1024 Mar 14 07:30 ..
drwxr-xr-x   2 adm      adm           512 Feb 13 12:15 acct
lrwxrwxrwx   1 root     root           14 Feb 13 10:37 aliases ->
./mail/aliases
drwxr-xr-x   3 root     bin           512 Feb 13 12:15 apache
-rw-r--r--   1 root     bin            50 Feb 13 10:40 auto_home
-rw-r--r--   1 root     bin           113 Feb 13 10:40 auto_master
lrwxrwxrwx   1 root     root           16 Feb 13 10:26 autopush ->
../sbin/autopush
dr-xr-xr-x   3 root     root          512 Feb 28 14:07 certs
lrwxrwxrwx   1 root     root           18 Feb 13 10:26 cfgadm ->
../usr/sbin/cfgadm
lrwxrwxrwx   1 root     root           18 Feb 13 10:35 chroot ->
../usr/sbin/chroot
lrwxrwxrwx   1 root     root           16 Feb 13 10:26 clri ->
../usr/sbin/clri
-rw-r--r--   1 root     other         314 Mar 14 07:06 coreadm.conf
-rw-r--r--   1 root     root         2236 Mar 14 07:06 .cpr_config
lrwxrwxrwx   1 root     root           16 Feb 13 10:26 cron ->
../usr/sbin/cron
(output truncated)
```

A long listing of a device file shows two numbers, separated by a comma. These two numbers are called major and minor device numbers. In the following example, the device file `dad@0,0:a` refers to major device number 136 and minor device number 0.

```
# cd /devices/pci@1f,0/pci@1,1/ide@3
# ls -l dad@0*
total 0
brw-r-----    1 root     sys       136,  0 Sep  6 15:40 dad@0,0:a
crw-r-----    1 root     sys       136,  0 Sep  6 15:40 dad@0,0:a,raw
brw-r-----    1 root     sys       136,  1 Sep  7 07:37 dad@0,0:b
```

```
crw-r-----   1 root     sys      136,  1 Sep  6 15:40 dad@0,0:b,raw
brw-r-----   1 root     sys      136,  2 Sep  6 15:40 dad@0,0:c
crw-r-----   1 root     sys      136,  2 Sep  6 15:40 dad@0,0:c,raw
brw-r-----   1 root     sys      136,  3 Sep  6 15:40 dad@0,0:d
crw-r-----   1 root     sys      136,  3 Sep  6 15:40 dad@0,0:d,raw
brw-r-----   1 root     sys      136,  4 Sep  6 15:40 dad@0,0:e
crw-r-----   1 root     sys      136,  4 Sep  6 15:40 dad@0,0:e,raw
brw-r-----   1 root     sys      136,  5 Sep  6 15:40 dad@0,0:f
crw-r-----   1 root     sys      136,  5 Sep  6 15:40 dad@0,0:f,raw
brw-r-----   1 root     sys      136,  6 Sep  6 15:40 dad@0,0:g
crw-r-----   1 root     sys      136,  6 Sep  6 15:40 dad@0,0:g,raw
brw-r-----   1 root     sys      136,  7 Sep  6 15:40 dad@0,0:h
crw-r-----   1 root     sys      136,  7 Sep  6 15:40 dad@0,0:h,raw
```

A major device number identifies the specific device driver required to access a device. A minor device number identifies the specific unit of the type that the device driver controls.

The device file `dad@0,0:a`, shown in Figure 1-6, occupies inode number 90681. The inode contains the major and minor device numbers that refer to a specific device, in this case, a slice on a disk.

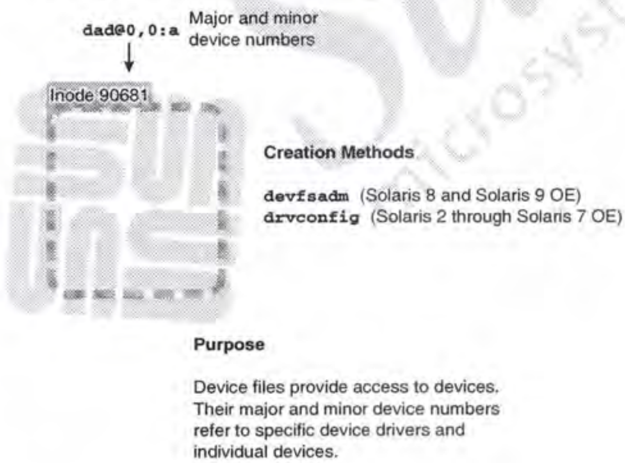

Figure 1-6 Device Files

In general, a reconfiguration boot creates device files and symbolic links to the device files automatically. In both the Solaris 8 OE and Solaris 9 OE, you can use the `devfsadm` command to create new device files manually.

A direct relationship exists between the device file and the device it controls. The major and minor device numbers contained in the inode establish this relationship.

Figure 1-7 shows the relationship between the device file dad@0,0:a and the disk device it controls. The inode information for dad@0,0:a contains major number 136 and minor number 0. Major number 136 identifies the dad device driver. The dad device driver controls integrated device electronics (IDE) disk drives. Minor number 0, in this case, identifies Slice 0.

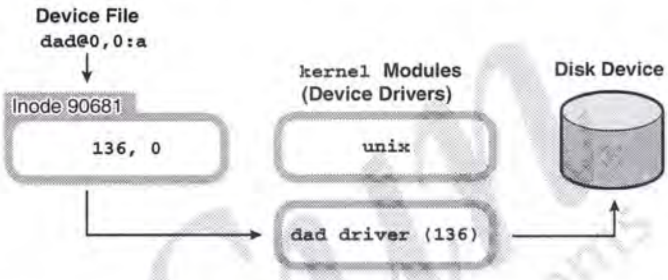

Figure 1-7 Device File Example

Device files fall into two categories: character-special devices and block-special devices. Character-special devices are also called character or raw devices. Block-special devices are often called block devices. Device files in these two categories interact with devices differently.

Character-Special Device Files

The file type "c" identifies character-special device files. For disk devices, character-special device files call for input/output (I/O) operations based on the disk's smallest addressable unit, a sector. Each sector is 512 bytes in size.

The following example shows a character-special device file.

```
crw-r-----   1 root     sys      136,  0 Feb 14 11:05 dad@0,0:a,raw
```

Block-Special Device Files

The file type "b" identifies block-special device files. For disk devices, block-special device files call for I/O operations based on a defined block size. The block size depends on the particular device, but for UNIX file systems (ufs), the default block size is 8 Kbytes.

The following example shows a block-special device file.

```
brw-r-----   1 root     sys      136,  0 Feb 14 11:05 dad@0,0:a
```

Using Hard Links

This section defines hard links and describes how to use them.

Introducing Hard Links

A hard link is the association between a file name and an inode. A hard link is not a separate type of file. Every type of file uses at least one hard link. Every entry in a directory constitutes a hard link. Think of every file name as a hard link to an inode. When you create a file, using the `touch` command, for example, a new directory entry is created that links the file name you specified with a particular inode. In this way, creating a new file creates a hard link.

In Figure 1-8, the file called `file1` is listed in the directory `dir1`. In `dir1`, the name `file1` is associated with inode number 1282. The hard link is the association between `file1` and inode number 1282.

Figure 1-8 Hard Link

Information in each inode keeps count of the number of file names associated with it. This is called a link count. In the output from the `ls -l` command, the link count appears between the column of file permissions and the column identifying the owner. In the following example, the file called `file1` uses one hard link.

```
# cd dir1
# touch file1
# ls -l
total 0
-rw-r--r--   1 root     other          0 Apr  7 15:26 file1
```

Creating New Hard Links

A new hard link for a file name increments the link count in the associated inode.

In the following example, inode 1282 now has two hard links, one for `file1` and the other for `file2`. The `ls -li` command lists the inode number in the left-most column. The `find -inum` command locates files and directories that have the same inode numbers.

```
# ln file1 file2
# ls -l
total 0
-rw-r--r--   2 root     other          0 Apr  7 15:26 file1
-rw-r--r--   2 root     other          0 Apr  7 15:26 file2
# ls -li
total 0
   1282 -rw-r--r--   2 root     other          0 Apr  7 15:26 file1
   1282 -rw-r--r--   2 root     other          0 Apr  7 15:26 file2
# find . -inum 1282
./file1
./file2
```

The `ln` command creates new hard links to regular files.

For example, the `ln file1 file2` command creates a new directory entry called `file2`. The `file2` file is associated with the same inode that is associated with `file1`.

Figure 1-9 shows the result of the `ln` command. Two file names are associated with inode number 1282. Unlike symbolic links, hard links cannot span file systems.

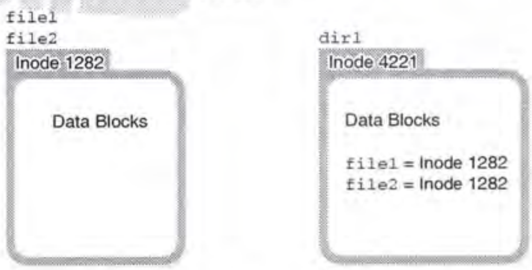

Figure 1-9 File Names Associated With an Inode Number

Removing Hard Links

Deleting one of the files has no effect on the other file. The link count decrements accordingly.

The following example shows how deleting `file1` from the previous example has no effect on `file2`.

```
# rm file1
# ls -li
total 0
1282 -rw-r--r--   1 root     other          0 Apr  7 15:26 file2
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Identifying File Types (Level 1)

In this exercise, you complete the following tasks:

- Navigate within the directory hierarchy
- Identify different types of files

Preparation

Refer to the lecture notes as necessary to perform the following steps and answer the following questions.

Tasks

Complete the following tasks:

- Identify the first symbolic link listed in the / (root) directory. Record the symbolic link's size and the name of the file it references. Identify the types of files found in the /dev/dsk directory and the types of files that the symbolic links reference, if any. Identify the types of files found in the /dev/pts directory and the types of files that the symbolic links reference, if any.

  (Steps 1–5 in Level 2 lab)

- Identify the types of files found in the /etc/init.d directory. Record the inode number and link count for the nfs.server file. Use the find command to locate all other files below the /etc directory that use the same inode as nfs.server.

  (Steps 6–8 in Level 2 lab)

- Create a directory called /testdir. In this directory, create a file and a symbolic link that points to the file. Determine if the two files use the same or a different inode.

  Create a directory called newdir within the /testdir directory. Identify the inode it uses, its link count, and the name of any other file that uses the same inode as the newdir directory.

  Create another directory below the newdir directory. Determine how the link count for the newdir directory changes, and find any new file that uses the same inode as the newdir directory.

  (Steps 9–14 in Level 2 lab)

Exercise: Identifying File Types (Level 2)

In this exercise, you complete the following tasks:

- Navigate within the directory hierarchy
- Identify different types of files

Preparation

Refer to the lecture notes as necessary to perform the following steps and answer the following questions.

Task Summary

Complete the following tasks:

- Identify the first symbolic link listed in the / (root) directory. Record the symbolic link's size and the name of the file it references. Identify the types of files found in the `/dev/dsk` directory and the types of files that the symbolic links reference, if any. Identify the types of files found in the `/dev/pts` directory and the types of files that the symbolic links reference, if any.
- Identify the types of files found in the `/etc/init.d` directory. Record the inode number and link count for the `nfs.server` file. Use the `find` command to locate all other files below the `/etc` directory that use the same inode as `nfs.server`.
- Create a directory called `/testdir`. In this directory, create a file and a symbolic link that points to the file. Determine if the two files use the same or a different inode.

  Create a directory called `newdir` within the `/testdir` directory. Identify the inode it uses, its link count, and the name of any other file that uses the same inode as the `newdir` directory.

  Create another directory below the `newdir` directory. Determine how the link count for the `newdir` directory changes, and find any new file that uses the same inode as the `newdir` directory.

Tasks

Complete the following steps:

1. Log in as the root user, and open a terminal window. In the / (root) directory, perform a long listing and record the name of the first symbolic link listed.
2. What is the size in bytes of the link you found in Step 1? How many characters are there in the name of the file to which this link points?
3. Change to the /dev/dsk directory. Record the file types that you find in this directory.
4. Use the appropriate options for the ls command to display information for the files that are referenced by the files in the /dev/dsk directory. Record the file types reported.
5. Change to the /dev/pts directory, and use the same commands you used in Steps 3 and 4 for the /dev/dsk directory. Record the file types you find.
6. Change to the /etc/init.d directory, and identify the type of file in this directory.
7. How many hard links are associated with the /etc/init.d/nfs.server file? What is the inode number associated with this file?
8. Find the number of files in the /etc directory or below that have the same inode number as that used by the /etc/init.d/nfs.server file.
9. Create a new directory called /testdir. Create a file in this directory called file1. Create a symbolic link called link1 that points to file1.
10. List file1 and the link1 symbolic link. Do these files use the same or different inodes?
11. In the /testdir directory, create a new directory called newdir. What is the number of hard links associated with the newdir directory? What is the inode number associated with the newdir directory?
12. List all files, including hidden files, that exist in the newdir directory. Which of these files uses the same inode as the newdir directory?

13. Create a new directory called dir2 below the newdir directory. What happens to the link count for the newdir directory?
14. Use the ls command with appropriate options to find the new file name that uses the same inode as the newdir directory. Record the name of the new file.

Exercise: Identifying File Types (Level 3)

In this exercise, you complete the following tasks:

- Navigate within the directory hierarchy
- Identify different types of files

Preparation

Refer to the lecture notes as necessary to perform the following steps and answer the following questions.

Task Summary

In this exercise, you accomplish the following:

- Identify the first symbolic link listed in the `/` (root) directory. Record the symbolic link's size and the name of the file it references. Identify the types of files found in the `/dev/dsk` directory and the types of files that the symbolic links reference, if any. Identify the types of files found in the `/dev/pts` directory and the types of files that the symbolic links reference, if any.
- Identify the types of files found in the `/etc/init.d` directory. Record the inode number and link count for the `nfs.server` file. Use the `find` command to locate all other files below the `/etc` directory that use the same inode as `nfs.server`.
- Create a directory called `/testdir`. In this directory, create a file and a symbolic link that points to the file. Determine if the two files use the same or a different inode.

  Create a directory called `newdir` within the `/testdir` directory. Identify the inode it uses, its link count, and the name of any other file that uses the same inode as the `newdir` directory.

  Create another directory below the `newdir` directory. Determine how the link count for the `newdir` directory changes, and find any new file that uses the same inode as the `newdir` directory.

Tasks and Solutions

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. In the / (root) directory, perform a long listing, and record the name of the first symbolic link listed.

```
# cd /
# ls -l
```

The `/bin` *symbolic link should be the first link listed in the / (root) directory.*

2. What is the size in bytes of the link you found in Step 1? How many characters are there in the name of the file to which this link points?

The `/bin` *symbolic link contains 9 bytes of data and points to* `./usr/bin`.

3. Change to the `/dev/dsk` directory. Record the file types that you find in this directory.

```
# cd /dev/dsk
# ls -l
```

The `/dev/dsk` *directory contains symbolic links.*

4. Use the appropriate options of the `ls` command to display information for the files referenced by the files in the `/dev/dsk` directory. Record the file types reported.

```
# ls -lL
```

The symbolic links in the `/dev/dsk` *directory point to block-special device files.*

5. Change to the `/dev/pts` directory, and use the same commands you used in Steps 3 and 4 for the `/dev/dsk` directory. Record the file types you find.

```
# cd /dev/pts
# ls -l
# ls -lL
```

The `/dev/pts` *directory contains symbolic links.*

The symbolic links in the `/dev/pts` *directory point to character-special device files.*

6. Change to the `/etc/init.d` directory, and identify the type of file in this directory.

```
# cd /etc/init.d ; ls -l
```

The `/etc/init.d` *directory contains regular files.*

7. How many hard links are associated with the `/etc/init.d/nfs.server` file? What is the inode number associated with this file?

```
# ls -li nfs.server
```

The `/etc/init.d/nfs.server` file has six hard links associated with it. The inode number varies among different systems.

8. Find the number of files in the `/etc` directory or below that have the same inode number as that used by the `/etc/init.d/nfs.server` file.

```
# ls -i /etc/init.d/nfs.server
   176603 /etc/init.d/nfs.server
# find /etc -inum 176603
```

Six files, including `nfs.server`, use the same inode number. They are:

```
/etc/init.d/nfs.server
/etc/rc0.d/K28nfs.server
/etc/rc1.d/K28nfs.server
/etc/rc2.d/K28nfs.server
/etc/rc3.d/S15nfs.server
/etc/rcS.d/K28nfs.server
```

9. Create a new directory called `/testdir`. Create a file in this directory called `file1`. Create a symbolic link called `link1` that points to `file1`.

```
# mkdir /testdir
# cd /testdir
# touch file1
# ln -s file1 link1
```

10. List `file1` and the `link1` symbolic link. Do these files use the same or different inodes?

```
# ls -li
```

These two files use two different inodes.

11. In the `/testdir` directory, create a new directory called `newdir`. What is the number of hard links associated with the `newdir` directory? What is the inode number associated with the `newdir` directory?

```
# mkdir newdir
# ls -ldi newdir
```

The link count for the `newdir` directory is two. The inode number varies among different systems.

12. List all files, including hidden files, that exist in the `newdir` directory. Which of these files uses the same inode as the `newdir` directory?

```
# ls -lia newdir
```

The file called dot (`.`) uses the same inode as the `newdir` *directory.*

13. Create a new directory called `dir2` below the `newdir` directory. What happens to the link count for the `newdir` directory?

```
# mkdir newdir/dir2
# ls -ldi newdir
```

The link count increases from two to three.

14. Use the `ls` command with appropriate options to find the new file name that uses the same inode as the `newdir` directory. Record the name of the new file.

```
# ls -laRi newdir
```

The `newdir/dir2/..` *file uses the same inode as the* `newdir` *directory.*

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 2

Managing Local Disk Devices

Objectives

Upon completion of this module, you should be able to:

- Describe the basic architecture of a disk
- Describe the naming conventions for devices
- List devices
- Reconfigure devices
- Perform hard disk partitioning
- Manage disk labels
- Describe the Solaris Management Console
- Partition a disk by using the Solaris Management Console

The following course map shows how this module fits into the current instructional goal.

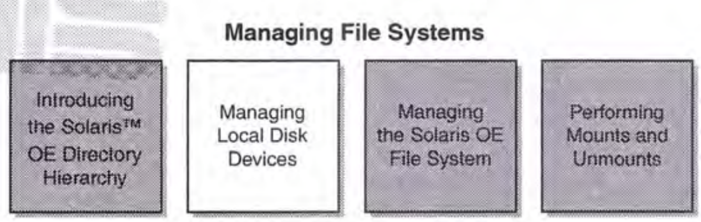

Figure 2-1 Course Map

Introducing the Basic Architecture of a Disk

A disk device has physical components and logical components. The physical components include disk platters and read/write heads. The logical components include disk slices, cylinders, tracks, and sectors.

Physical Disk Structure

A disk is physically composed of a series of flat, magnetically coated platters that are stacked on a spindle. The spindle turns while the read/write heads move as a single unit radially, reading and writing data on the platters.

Figure 2-2 identifies the parts of a disk.

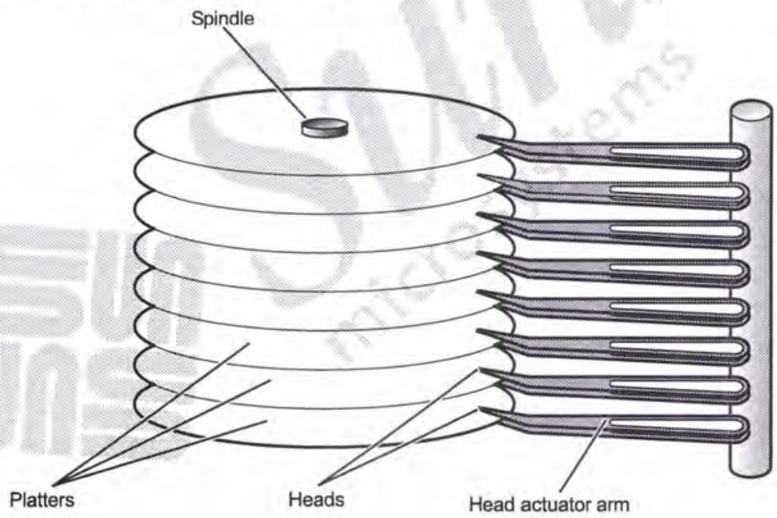

Figure 2-2 Components of a Disk

The following list describes the physical components of a disk:

- The disk storage area is composed of one or more platters.
- The platters rotate.
- The head actuator arm moves the read/write heads as a unit radially.
- The read/write heads read and write data on the magnetic surface on both sides of the platters.

Data Organization on Disk Platters

Figure 2-3 shows the logical components of a disk platter.

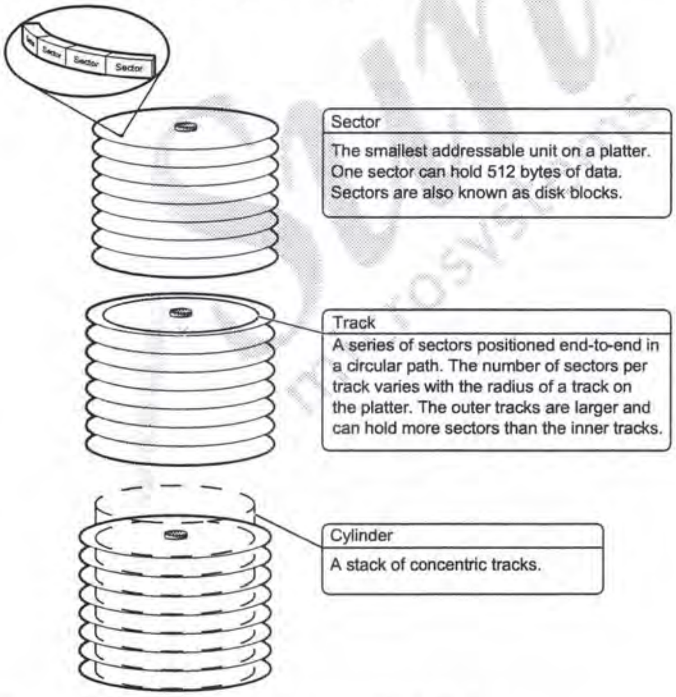

Figure 2-3 Data Organization on Disk Platters

A disk platter is divided into sectors, tracks, and cylinders.

Sector	The smallest addressable unit on a platter. One sector can hold 512 bytes of data. Sectors are also known as disk blocks.
Track	A series of sectors positioned end-to-end in a circular path.
Cylinder	A stack of tracks.

The number of sectors per track varies with the radius of a track on the platter. The outermost tracks are larger and can hold more sectors than the inner tracks.

Because a disk spins continuously and the read/write heads move as a single unit, the most efficient seeking occurs when the sectors to be read from or written to are located in a single cylinder.

Disk Slices

Disks are logically divided into individual partitions known as disk slices. Disk slices are groupings of cylinders that are commonly used to organize data by function.

For example, one slice can store critical system files and programs while another slice on the same disk can store user-created files.

Note – Grouping cylinders into slices is done to organize data, facilitate backups, and provide swap space.

A disk under the Solaris OE can be divided into eight slices that are labeled Slice 0 through Slice 7.

By convention, Slice 2 represents the entire disk. Slice 2 maintains important data about the entire disk, such as the size of the actual disk and the total number of cylinders available for the storage of files and directories.

A starting cylinder and an ending cylinder define each slice. These cylinder boundaries determine the size of a slice.

Figure 2-4 shows how disk slices might reside on a disk.

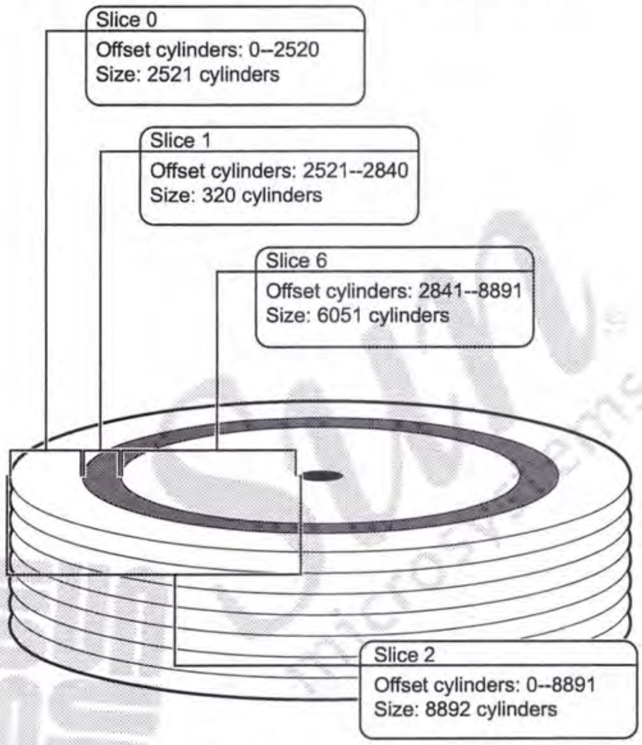

Figure 2-4 Cylinders and Slices

Table 2-1 shows disk slices and the different file systems they could hold.

Table 2-1 Disk Slices

Slice	Name	Function
0	/	The root directory's system files
1	swap	Swap area
2		Entire disk
5	/opt	Optional software
6	/usr	System executables and programs
7	/export/home	User files and directories

Figure 2-5 shows a possible configuration convention for organizing data. The example disk is divided into slices that logically organize the data on the boot disk.

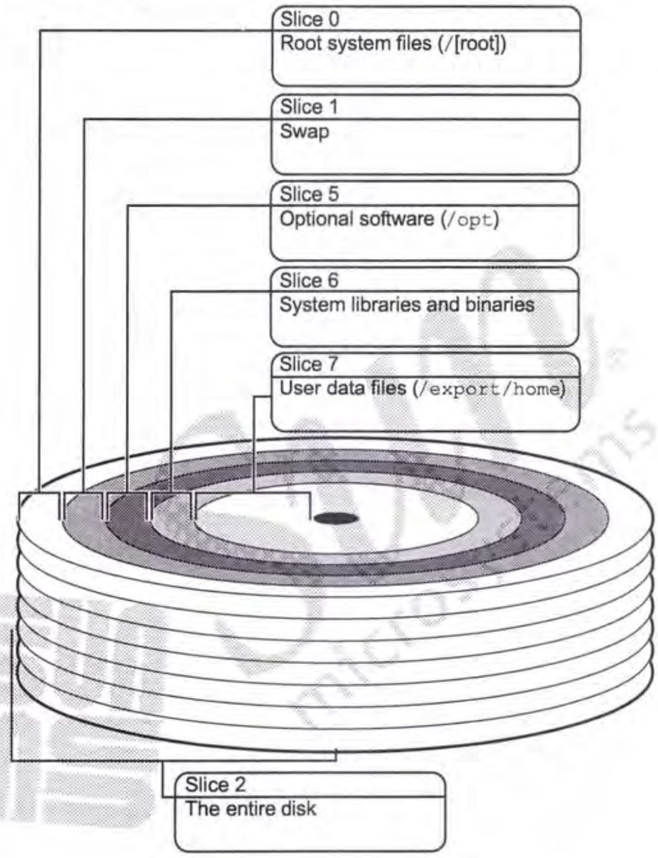

Figure 2-5 Top View of Five Configured Disk Slices

Disk Slice Naming Convention

An eight-character string typically represents the full name of a slice. The string includes the controller number, the target number, the disk number, and the slice number.

Controller number	Identifies the host bus adapter (HBA), which controls communications between the system and disk unit. The HBA takes care of sending and receiving both commands and data to the device. The controller number is assigned in sequential order, such as `c0`, `c1`, `c2`, and so on.
Target number	Target numbers, such as `t0`, `t1`, `t2`, and `t3`, correspond to a unique hardware address that is assigned to each disk, tape, or CD-ROM. Some external disk drives have an address switch located on the rear panel. Some internal disks have address pins that are jumpered to assign that disk's target number.
Disk number	The disk number is also known as the logical unit number (LUN). This number reflects the number of disks at the target location.
Slice number	A slice number ranging from 0 to 7.

The embedded SCSI configuration and the integrated device electronics (IDE) configuration represent the disk slice naming conventions across two different architectures. The disk number is always set to `d0` with embedded SCSI disks.

Figure 2-6 shows the eight-character string that represents the full name of a disk slice.

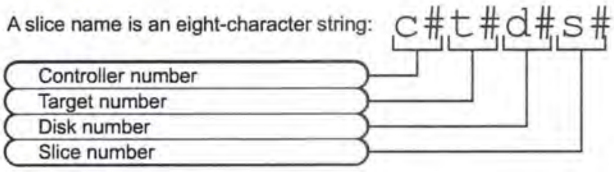

Figure 2-6 Disk Slice Naming Conventions

Figure 2-7 shows the configuration of the SCSI architecture.

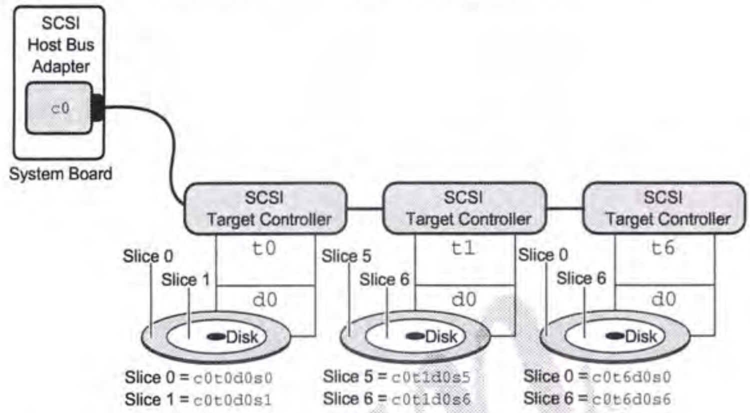

Figure 2-7 Embedded SCSI Configuration

Figure 2-8 shows the configuration of the IDE architecture.

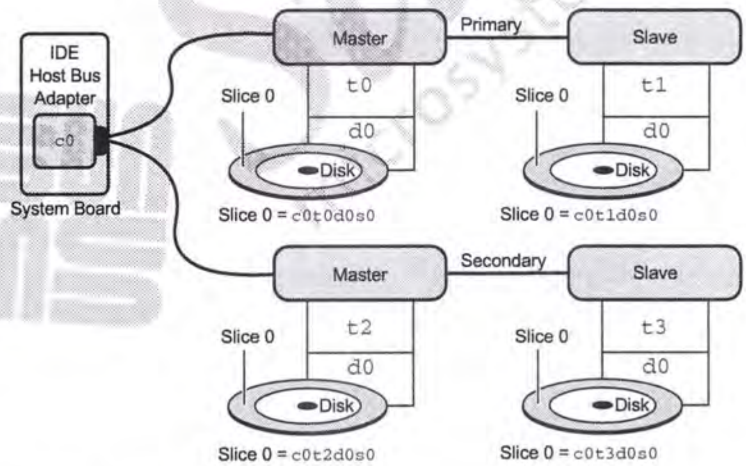

Figure 2-8 IDE Configuration

New I/O boards include on-board gigabit interface converter (GBIC) sockets that support the 100 Mbytes per second Fibre Channel Arbitrated Loop (FC-AL) connectivity. FC-AL typically uses glass fibre passing a light amplification of stimulated emission of radiation (laser) beam signals modulated with information. FC-AL can also use conventional electrical (copper) connections, such as a wire cable or backplane. These interfaces are typically found on Sun™ Enterprise-level servers. Naming conventions appear basically the same to the user.

Introducing Solaris OE Device Naming Conventions

In the Solaris OE, all devices are represented by three different types of names, depending on how the device is being referenced:

- Logical device names
- Physical device names
- Instance names

Note – The Berkeley Software Distribution (BSD) device names exist in the Solaris OE if the BSD compatibility packages are installed with the Solaris Developer, Entire Distribution, or Entire Distribution Plus the original equipment manufacturer (OEM) Solaris software group. The BSD device names, for example `/dev/sd0a`, are typically used for backward compatibility with old scripts.

Logical Device Names

Logical disk device names are symbolic links to the physical device names kept in the `/devices` directory. Logical device names are used primarily to refer to a device when you are entering commands on the command line. All logical device names are kept in the `/dev` directory. The logical device names contain the controller number, target number, disk number, and slice number.

Every disk device has an entry in both the `/dev/dsk` and `/dev/rdsk` directories for the block and character disk devices, respectively. To display the entries in the `/dev/dsk` directory, perform the command:

```
# ls /dev/dsk
c0t0d0s0  c0t0d0s4  c0t2d0s0  c0t2d0s4  c1t1d0s0  c1t1d0s4
c0t0d0s1  c0t0d0s5  c0t2d0s1  c0t2d0s5  c1t1d0s1  c1t1d0s5
c0t0d0s2  c0t0d0s6  c0t2d0s2  c0t2d0s6  c1t1d0s2  c1t1d0s6
c0t0d0s3  c0t0d0s7  c0t2d0s3  c0t2d0s7  c1t1d0s3  c1t1d0s7
```

- `c0t0d0s0` through `c0t0d0s7` – Identifies the device names for disk Slices 0 through 7 for a disk that is attached to Controller 0, at Target 0, on Disk Unit 0.

- `c0t2d0s0` through `c0t2d0s7` – Identifies the device names for disk Slices 0 through 7 for a disk that is attached to Controller 0, at Target 2, on Disk Unit 0.
- `c1t1d0s0` through `c1t1d0s7` – Identifies the device names for disk Slices 0 through 7 for a disk that is attached to Controller 1, at Target 1, on Disk Unit 0.

Physical Device Names

Physical device names uniquely identify the physical location of the hardware devices on the system and are maintained in the `/devices` directory.

A physical device name contains the hardware information, represented as a series of node names, separated by slashes, that indicate the path to the device. To display a physical device name, perform the command:

```
# ls -l /dev/dsk/c0t0d0s0
lrwxrwxrwx   1 root     root 46 Jun 16 19:07 /dev/dsk/c0t0d0s0 ->
../../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a
```

FC-AL disks will appear slightly different because they also display a World Wide Name (WWN). The following example was taken from a Sun™ Enterprise 3500 server:

```
# ls -l /dev/rdsk/c0t0d0s0
lrwxrwxrwx 1 root root 78 Jun 16 2000 /dev/rdsk/c0t0d0s0 ->
../../devices/sbus@2,0/SUNW,socal@d,10000/sf@0,0/ssd@w21000020375b9ab6,0:
a,raw
```

Figure 2-9 shows the device configuration hierarchy of an Ultra 5 workstation. Not all possible devices are included.

```
/devices
└─pci@1f
  └─pci@1,1
    ├─ebus@1
    │ ├─fdthree@14,
    │ └─se@14,
    └─ide@3
      ├─dad@0,0
      └─sd@2,0
```

Figure 2-9 The `/devices` Directory Structure

Note – Various hardware platforms have different device trees.

The top-most directory in the hierarchy is called the root node of the device tree. The bus nexus nodes and the leaf nodes below the root object have device drivers associated with them.

A device driver is the software that communicates with the device. This software must be available to the `kernel` so that the system can use the device.

During system initialization, the `kernel` identifies the physical location of a device. The `kernel` associates a node with an address, *`nodename@address`*, which is the physical device name.

In Figure 2-9, `dad@0` is the direct access disk device at address 0.

Instance Names

Instance names are abbreviated names assigned by the `kernel` for each device on the system.

An instance name is a shortened name for the physical device name. Two examples are shown:

- `sd`*n*

  where `sd` is the disk name and *n* is the number, such as `sd0` for the first SCSI disk device.

- `dad`*n*

  where `dad` (direct access device) is the disk name and *n* is the number, such as `dad0` for the first IDE disk device.

Listing a System's Devices

In the Solaris OE, there are several ways to list a system's devices, including:

- Using the `/etc/path_to_inst` file
- Using the `prtconf` command
- Using the `format` command

The `/etc/path_to_inst` File

For each device, the system records its physical name and instance name in the `/etc/path_to_inst` file. These names are used by the `kernel` to identify every possible device. This file is read only at boot time.

The `/etc/path_to_inst` file is maintained by the `kernel`, and it is generally not necessary, nor is it advisable, for the system administrator to change this file.

The following example shows entries in the `/etc/path_to_inst` file. The text within the parentheses indicates what device is referred to by the entry and does not appear in the actual file.

```
# more /etc/path_to_inst
#
#       Caution! This file contains critical kernel state
(output edited for clarity)
"/pci@1f,0" 0 "pcipsy" (PCI bus controller, "psycho" chip)
"/pci@1f,0/pci@1,1" 0 "simba" (PCI bus B)
"/pci@1f,0/pci@1,1/ebus@1" 0 "ebus" (extended bus)
"/pci@1f,0/pci@1,1/ebus@1/power@14,724000" 0 "power" (power management bus)
"/pci@1f,0/pci@1,1/ebus@1/fdthree@14,3023f0" 0 "fd" (floppy disk)
"/pci@1f,0/pci@1,1/ebus@1/SUNW,CS4231@14,200000" 0 "audiocs" (crystal
semiconductor)
"/pci@1f,0/pci@1,1/ebus@1/su@14,3062f8" 1 "su" (mouse)
"/pci@1f,0/pci@1,1/ebus@1/se@14,400000" 0 "se" (serial ports A and B)
"/pci@1f,0/pci@1,1/ebus@1/su@14,3083f8" 0 "su" (keyboard)
"/pci@1f,0/pci@1,1/ebus@1/ecpp@14,3043bc" 0 "ecpp" (extended capability
parallel port)
"/pci@1f,0/pci@1,1/ide@3" 0 "uata" (ATA controller)
"/pci@1f,0/pci@1,1/ide@3/sd@2,0" 0 "sd" (CD-ROM)
"/pci@1f,0/pci@1,1/ide@3/dad@0,0" 0 "dad" (hard disk)
"/pci@1f,0/pci@1,1/network@1,1" 0 "hme" (Fast Ethernet)
```

```
"/pci@1f,0/pci@1,1/SUNW,m64B@2" 0 "m64" (color memory frame buffer)
"/pci@1f,0/pci@1" 1 "simba" (PCI bus A controller)
"/options" 0 "options"
"/scsi_vhci" 0 "scsi_vhci"
"/pseudo" 0 "pseudo"
```

The device instance number, shown in the preceding example, appears to the left of the device instance name when recorded in this file.

Note – Different systems have different physical device paths. The preceding example shows an on-board peripheral component interconnect (PCI) bus configuration.

The following is a /etc/path_to_inst file from a system that has a different bus architecture. In this case, it is an example of a system that has an on-board Sun System bus (SBus).

```
# more /etc/path_to_inst
#
#       Caution! This file contains critical kernel state
#
"/sbus@1f,0" 0 "sbus"
"/sbus@1f,0/espdma@e,8400000" 0 "dma"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000" 0 "esp"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@3,0" 3 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@2,0" 2 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@1,0" 1 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@0,0" 0 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@6,0" 6 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@5,0" 5 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@4,0" 4 "sd"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@3,0" 3 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@2,0" 2 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@1,0" 1 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@0,0" 0 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@6,0" 6 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@5,0" 5 "st"
"/sbus@1f,0/espdma@e,8400000/esp@e,8800000/st@4,0" 4 "st"
... < remaining lines removed > ...
```

The following example is an /etc/path_to_inst file with an FC-AL entry:

```
"/sbus@2,0/SUNW,socal@d,10000/sf@0,0/ssd@w21000020375b9ab6,0" 0 "ssd"
```

The prtconf Command

Use the prtconf command to display the system's configuration information, including the total amount of memory installed and the configuration of system peripherals, which is formatted as a device tree.

The prtconf command lists all possible instances of devices, whether the device is attached or not attached to the system. To view a list of only attached devices on the system, perform the command:

```
# prtconf | grep -v not
System Configuration:  Sun Microsystems  sun4u
Memory size: 128 Megabytes
System Peripherals (Software Nodes):

SUNW,Ultra-5_10
    options, instance #0
    pci, instance #0
        pci, instance #0
            ebus, instance #0
                power, instance #0
                se, instance #0
                su, instance #0
                su, instance #1
                fdthree, instance #0
            network, instance #0
            SUNW,m64B, instance #0
            ide, instance #0
                dad, instance #0
                sd, instance #0
        pci, instance #1
            scsi, instance #0
            scsi, instance #1
    pseudo, instance #0
#
```

Note – The grep -v not command is used to omit all lines containing the word "not" from the output (such as driver not attached).

The `format` Command

Use the `format` command to display both logical and physical device names for all currently available disks. To view the logical and physical devices for currently available disks, perform the command:

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
          /pci@1f,0/pci@1,1/ide@e/dad@0,0
       1. c1t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
          /pci@1f,0/pci@1/SUNW,isptwo@4/sd@0,0
Specify disk (enter its number):
```

Note – Press Control-D to exit the `format` command.

Reconfiguring Devices

The system recognizes a newly added peripheral device if a reconfiguration boot is invoked or if the devfsadm command is run.

Performing a Reconfiguration Boot

For example, you can use a boot process to add a new device to a newly generated /etc/path_to_inst file and to the /dev and /devices directories.

The following steps reconfigure a system to recognize a newly attached disk.

1. Create the /reconfigure file. This file causes the system to check for the presence of any newly installed devices the next time it is powered on or booted.

```
# touch /reconfigure
#
```

2. Shut down the system by using the init 5 command. This command safely powers off the system, allowing for addition or removal of devices. (If the device is already attached to your system, you can shut down to the ok prompt with the command init 0.)

```
# init 5
```

3. Turn off the power to all external devices.
4. Install the peripheral device. Make sure that the address of the device being added does not conflict with the address of other devices on the system.
5. Turn on the power to all external devices.
6. Turn on the power to the system. The system boots to the login window.
7. Verify that the peripheral device has been added by issuing either the prtconf or format command.

After the disk is recognized by the system, begin the process of defining disk slices.

Note – If the /reconfigure file was not created before the system was shut down, you can invoke a manual reconfiguration boot with the programmable read-only memory (PROM) level command: boot -r.

Using the devfsadm Command

Many systems are running critical customer applications on a 24-hour, 7-day-a-week basis. It might not be possible to perform a reconfiguration boot on these systems. In this situation, you can use the devfsadm command.

The devfsadm command performs the device reconfiguration process and updates the /etc/path_to_inst file and the /dev and /devices directories during reconfiguration events.

The devfsadm command attempts to load every driver in the system and attach all possible device instances. It then creates the device files in the /devices directory and the logical links in the /dev directory. In addition to managing these directories, the devfsadm command also maintains the /etc/path_to_inst file.

To restrict the operation of the devfsadm command to a specific device class, use the -c option.

```
devfsadm -c device_class
```

The values for *device_class* include disk, tape, port, audio, and pseudo. For example, to restrict the devfsadm command to the disk device class, perform the command:

```
# devfsadm -c disk
```

Use the -c option more than once on the command line to specify multiple device classes. For example, to specify the disk, tape, and audio device classes, perform the command:

```
# devfsadm -c disk -c tape -c audio
```

To restrict the use of the devfsadm command to configure only devices for a named driver, use the -i option.

```
devfsadm -i driver_name
```

The following examples use the `-i` option.

- To configure only those disks supported by the `dad` driver, perform the command:

```
# devfsadm -i dad
```

- To configure only those disks supported by the `sd` driver, perform the command:

```
# devfsadm -i sd
```

- To configure devices supported by the `st` driver, perform the command:

```
# devfsadm -i st
```

To print the changes made by the `devfsadm` command to the `/dev` and `/devices` directories, perform the command:

```
# devfsadm -v
```

To invoke cleanup routines that remove unreferenced symbolic links for devices, perform the command:

```
# devfsadm -C
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab is more difficult. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Configuring and Naming Devices (Level 1)

In this exercise, you complete the following tasks:

- Identify logical, physical, and instance names for disk devices
- View the /etc/path_to_inst file for information about your boot disk
- Add a new disk or tape drive to a system
- Create new device files for the new disk or tape

Preparation

This exercise requires a system that is configured with an external disk or tape drive. During system boot, this external disk must remain powered off to avoid creating links and device files.

Tasks

Complete the following tasks:

- Identify the logical device name of your boot disk. Locate the logical device files in the /dev/dsk and /dev/rdsk directories for Slice 0 on this disk, and record their true file types.
- Locate the physical device names that are associated with both logical device names that you found for your boot disk. Record their true file types. (Steps 1–5 in the Level 2 lab)
- In the /etc/path_to_inst file, identify and record the instance name for your boot disk. (Steps 6–7 in the Level 2 lab)
- Confirm that no links or device files exist for the disk or tape device that you want to connect. Halt the system, and power on the device. Boot the system to its default run state. Run the devfsadm command in verbose mode to create new links and device files, and check the directories in which you created them to confirm that they exist. (Task 1, Steps 1–2, and Task 2, Steps 1–4, in the Level 2 lab)

Exercise: Configuring and Naming Devices (Level 2)

In this exercise, you complete the following tasks:

- Identify logical, physical, and instance names for disk devices
- View the /etc/path_to_inst file for information about your boot disk
- Add a new disk or tape drive to a system
- Create new device files for the new disk or tape

Preparation

This exercise requires a system that is configured with an external disk or tape drive. During system boot, this external disk must remain powered off to avoid creating links and device files.

Task Summary

Complete the following tasks:

- Identify the logical device name of your boot disk. Locate the logical device files in the /dev/dsk and /dev/rdsk directories for Slice 0 on this disk, and record their true file types.
- Locate the physical device names that are associated with both logical device names that you found for your boot disk. Record their true file types.
- In the /etc/path_to_inst file, identify and record the instance name for your boot disk.
- Confirm that no links or device files exist for the disk or tape device that you want to connect. Halt the system, and power on the device. Boot the system to its default run state. Run the devfsadm command in verbose mode to create new links and device files, and check the directories in which you created them to confirm that they exist.

Tasks

Complete the following tasks.

Task 1 – Identifying Device Files

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Expand the window so that it occupies the entire screen area. Change to the `/dev/dsk` directory.
2. List the files in this directory. Identify the files related to the boot disk of your system. Most systems use `c0t0d0`. Locate the item related to Slice 0 on this disk, and display a long listing of it.

   Which type of file did you just locate? The file type indicator is the first character on the left side of the long listing.

   Record the full path name to which this file points.
3. Highlight the path name you recorded in Step 2 by double-clicking the path name. Use the Copy and Paste keys to copy and paste this path name into a long listing command. If you are not using the CDE, you need to type the path name.

   Which type of file is this?

   The command `ls -lL c0t0d0s0` displays the same information but shows only the link file name, not the real device file name.
4. Change to the `/dev/rdsk` directory. Display a long listing of the same file name you selected in Step 2.

   Which type of file is this?

   Record the full path name to which this file points.
5. Highlight the path name you recorded in Step 4. Use the Copy and Paste keys to copy and paste this path name into a long listing command. If you are not using the CDE, you need to type in the path name.

   Which type of file is this?

   The `ls -lL c0t0d0s0` command displays the same information but shows only the link file name, not the real device file name.

6. Change the directory to the /etc directory. Display the contents of the path_to_inst file.
7. Use the information from the previous steps to locate and record the entry for your boot disk. An Ultra 5 workstation, for example, would use c0t0d0 as its boot disk. This relates to the device file called dad@0,0 and is listed in the /etc/path_to_inst directory.

   The instance name is composed of the dad or sd tag and the number that precedes it in the /etc/path_to_inst file. What is the instance name for the device listed in this step?

Task 2 – Adding a New Disk or Tape Device

Complete the following steps:

1. In the /dev/dsk and /dev/rmt directories, confirm that no files exist for your external disk or tape device, for example, /dev/dsk/c1t0d0s0 or /dev/rmt/0. If files for the external device do exist, ask your instructor to provide directions to remove them.
2. Shut down your system to run state 0.
3. Power on the external disk or tape drive attached to your system.
4. Boot the system to its default run state.
5. Log in as the root user, and open a terminal window. Run the devfsadm command with the -v option to create new links and device files for the new disk or tape drive. Observe the messages that the devfsadm command displays.
6. Confirm that new links and device files exist in the /dev/dsk and /dev/rdsk directories for disks or /dev/rmt for tape drives.

Exercise: Configuring and Naming Devices (Level 3)

In this exercise, you complete the following tasks:

- Identify logical, physical, and instance names for disk devices
- View the `/etc/path_to_inst` file for information about your boot disk
- Add a new disk or tape drive to a system
- Create new device files for the new disk or tape

Preparation

This exercise requires a system that is configured with an external disk or tape drive. During system boot, this external disk must remain powered off to avoid creating links and device files.

Task Summary

Complete the following tasks:

- Identify the logical device name of your boot disk. Locate the logical device files in the `/dev/dsk` and `/dev/rdsk` directories for Slice 0 on this disk, and record their true file types.
- Locate the physical device names that are associated with both logical device names that you found for your boot disk. Record their true file types.
- In the `/etc/path_to_inst` file, identify and record the instance name for your boot disk.
- Confirm that no links or device files exist for the disk or tape device that you want to connect. Halt the system, and power on the device. Boot the system to its default run state. Run the `devfsadm` command in verbose mode to create new links and device files, and check the directories in which you created them to confirm that they exist.

Tasks and Solutions

Complete the following tasks.

Task 1 – Identifying Device Files

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Expand the window so that it occupies the entire screen area. Change to the `/dev/dsk` directory.

```
# cd /dev/dsk
```

2. List the files in this directory. Identify the files related to the boot disk of your system. Most systems use `c0t0d0`. Locate the item related to Slice 0 on this disk, and display a long listing of it.

```
# ls
# ls -l c0t0d0s0
```

Which type of file did you just locate? The file type indicator is the first character on the left side of the long listing.

Files in this directory are symbolic links. The letter `l` *in the left-most column identifies a symbolic link.*

Record the full path name to which this file points.

Systems that use PCI bus architectures list path names similar to the following:

```
../../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a
```

Systems that use SBus architectures list path names similar to the following:

```
../../devices/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,
800000/sd@3,0:a
```

3. Highlight the path name you recorded in Step 2 by double-clicking the path name. Copy and paste this path name into a long listing command. If you are not using the CDE, you need to type the path name.

```
# ls -l pathname
```

Which type of file is this?

Files in this directory are device files. The `b` *character in the left-most column identifies a block-special device file.*

The command `ls -lL c0t0d0s0` displays the same information but shows only the link file name, not the real device file name.

4. Change to the /dev/rdsk directory. Display a long listing of the same file name you selected in Step 2.

```
# cd /dev/rdsk
# ls -l c0t0d0s0
```

Which type of file is this?

Files in this directory are symbolic links. The letter l *in the left-most column identifies a symbolic link.*

Record the full path name to which this file points.

Systems that use PCI bus architectures list path names similar to the following:

```
./../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a,raw
```

Systems that use SBus architectures list path names similar to the following:

```
../../devices/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,8000
00/sd@3,0:a,raw
```

5. Highlight the path name you recorded in Step 4. Use the Copy and Paste keys to copy and paste this path name into a long listing command. If you are not using CDE, you need to type in the path name.

```
# ls -l pathname
```

Which type of file is this?

Files in this directory are device files. The c *character in the left-most column identifies a character-special device file.*

The ls -lL c0t0d0s0 command displays the same information but shows only the link file name, not the real device file name.

6. Change to the /etc directory. Display the contents of the path_to_inst file.

```
# cd /etc
# more path_to_inst
```

7. Use the information from the previous steps to locate and record the entry for your boot disk. An Ultra 5 workstation, for example, would use c0t0d0 as its boot disk. This relates to the device file called dad@0,0 and is listed in the /etc/path_to_inst directory.

Systems that use PCI bus architectures list path names similar to the following:

```
/pci@1f,0/pci@1,1/ide@3/dad@0,0
```

Systems that use SBus architectures list path names similar to the following:

```
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
```

The instance name is composed of the `dad` or `sd` tag and the number that precedes it in the `/etc/path_to_inst` file. What is the instance name for the device listed in this step?

`dad0`, `sd3`, *or* `sd0`, *depending on the system architecture.*

Task 2 – Adding a New Disk or Tape Device

Complete the following steps:

1. In the `/dev/dsk` and `/dev/rmt` directories, confirm that no files exist for your external disk or device, for example, `/dev/dsk/c1t0d0s0` or `/dev/rmt/0`. If files for the external device do exist, ask your instructor to provide directions to remove them.
2. Shut down your system to run state 0.

```
# init 0
```

3. Power on the external disk or tape drive attached to your system.
4. Boot the system to its default run state.

```
ok boot
```

5. Log in as the `root` user, and open a terminal window. Run the `devfsadm` command with the `-v` option to create new links and device files for the new disk or tape drive. Observe the messages that the `devfsadm` command displays.

```
# devfsadm -v
```

6. Confirm that new links and device files exist in the `/dev/dsk` and `/dev/rdsk` directories for disks or `/dev/rmt` for tape drives.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Partitioning the Hard Disk

The `format` utility is a system administration tool used primarily to prepare hard disk drives for use in the Solaris OE.

Although the `format` utility also performs a variety of disk-management activities, the main function of the `format` utility is to divide a disk into disk slices.

Note – You do not need to partition the disk before you install the Solaris OE.

Introducing the Fundamentals of Disk Partitioning

To divide a disk into slices:

1. Identify the correct disk.
2. Plan the layout of the disk.
3. Use the `format` utility to divide the disk into slices.
4. Label the disk with new slice information.

Only the `root` user can use the `format` utility. If a regular user runs the `format` utility, the following error message appears:

```
$ format
 Searching for disk...done
 No permission (or no disk found)!
```

Recognizing Disk Space and Undesirable Conditions

Disk slices are defined by an offset and a size in cylinders. The offset is the distance from Cylinder 0. Figure 2-10 shows an example of disk slice sizes and offsets.

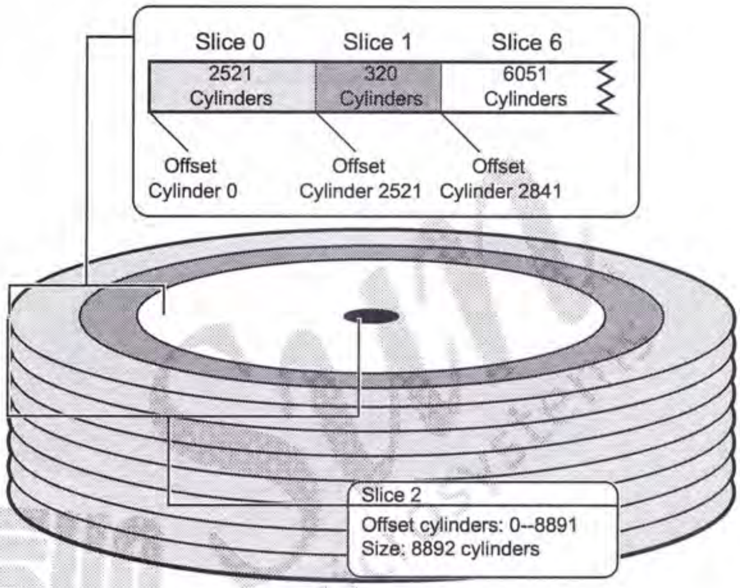

Figure 2-10 Offsets and Sizes for Disk Partitions

The offset for Slice 0 is 0 cylinders, and its size is 2521 cylinders. Slice 0 begins on Cylinder 0 and ends on Cylinder 2520.

The offset for Slice 1 is 2521 cylinders, and its size is 320 cylinders. Slice 1 begins on Cylinder 2521 and ends on Cylinder 2840.

The offset for Slice 6 is 2841 cylinders, and its size is 6051 cylinders. Slice 6 begins on Cylinder 2841 and ends on the last available cylinder, which is Cylinder 8891.

Recognizing Wasted Disk Space

Wasted disk space occurs when one or more cylinders are not allocated to a disk slice. Figure 2-11 shows a disk with cylinders that are not allocated.

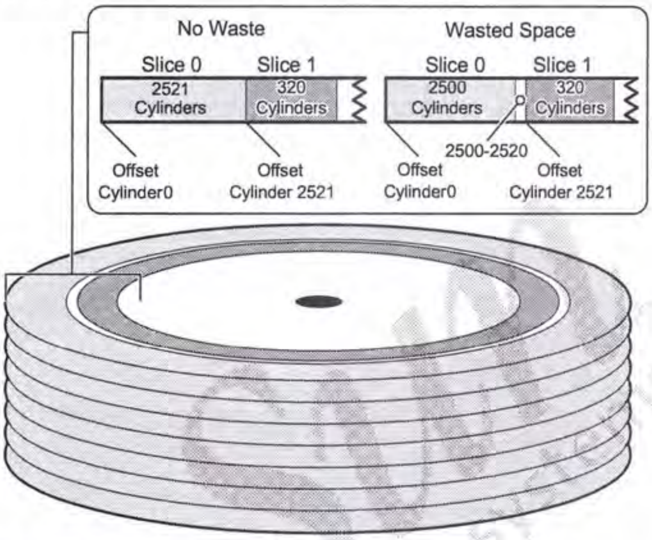

Figure 2-11 A Disk Slice With Wasted Space

Because the cylinders are not allocated to the disk slice, Cylinders 2500 through 2520 are unusable.

Wasted disk space occurs during partitioning when one or more cylinders have not been allocated to a disk slice. This may happen intentionally or accidentally. If there are unallocated slices available, then wasted space can possibly be assigned to a slice later on.

Recognizing Overlapping Disk Slices

Overlapping disk slices occur when one or more cylinders are allocated to more than one disk slice. Figure 2-12 shows a disk with cylinders allocated to more than one disk slice.

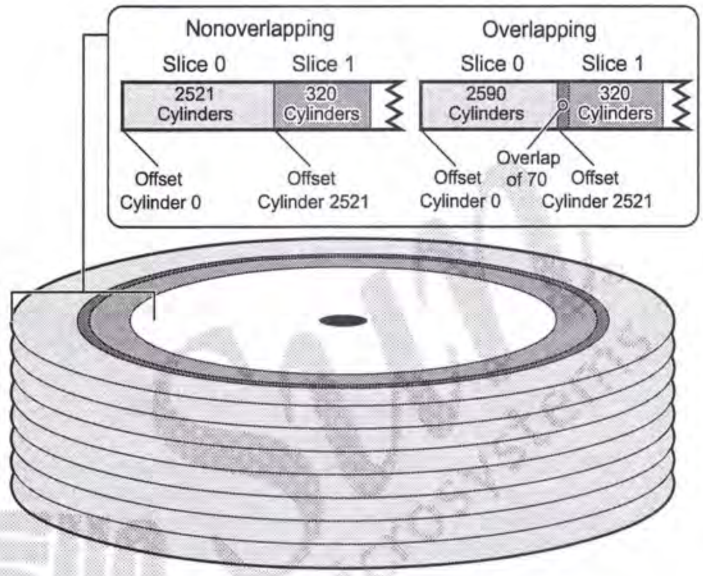

Figure 2-12 Disk Slices With Overlapping Cylinders

In Figure 2-12, Cylinders 2521 through 2590 are overlapping two disk slices.

This type of condition occurs when the size of one disk slice is increased and the starting cylinder number of the next disk slice is not adjusted. Only the `format` utility's `modify` command warns you of overlapping disk slices.

```
partition> modify
Select partitioning base:
        0. Current partition table (unnamed)
        1. All Free Hog
Choose base (enter number) [0]? 0
Warning: Overlapping partition (1) in table.
Warning: Fix, or select a different partition table.
```

Caution – Do not change the size of disk slices that are currently in use. When a disk with existing slices is repartitioned and relabeled, any existing data can become inaccessible. Copy existing data to backup media before the disk is repartitioned, and restore the data to the disk after the disk is relabeled and contains a new file system.

Introducing Disk Partition Tables

As the `root` user, when you use the `format` utility and select a disk to partition, a copy of the disk's partition table is read from the label on the disk into memory and is displayed as the current disk partition table.

The `format` utility also works with a file called `/etc/format.dat`, which is read when you invoke the `format` utility.

The `/etc/format.dat` file is a table of available disk types and a set of predefined partition tables that you can use to partition a disk quickly.

Introducing Disk Labels

The disk's label is the area set aside for storing information about the disk's controller, geometry, and slices. Another term used to describe a disk label is the volume table of contents (VTOC). The disk's label or VTOC is stored on the first sector of the disk.

To label a disk means to write slice information onto the disk. If you fail to label a disk after defining slices, the slice information is lost.

An important part of the disk label is the partition table, which identifies a disk's slices, the slice boundaries in cylinders, and the total size of the slices.

Note – The terms disk slice and disk partition are interchangeable.

Figure 2-13 shows the relationship among the label on the disk, the current label in memory, and the predefined label in the /etc/format.dat file.

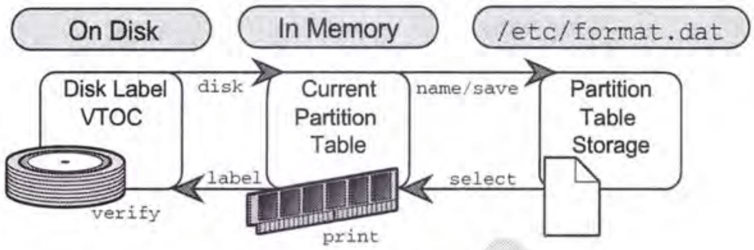

Figure 2-13 Partition Table Locations

Using the format Command

The format utility is organized into two tiers of commands.

When you type format on the command line, the first tier of commands appears. This set of commands allow you to, among other functions, select a disk, select a partition, save new disk and partition definitions, and write the label to the disk. The top tier of commands is denoted by the format> prompt.

The second tier of commands appears when you type partition from the format> prompt. This set of commands allow you to, among other functions, define the characteristics of the individual slices, print the existing partition table, and write the partition map and label to the disk.

Table 2-2 describes the terminology for disk partitioning.

Table 2-2 Partition Table Terms and Usage

Term	Description
Part	The slice number. Valid slice numbers are 0 through 7.
Tag	A value that indicates how the slice is being used. 0 = unassigned 1 = boot 2 = root 3 = swap 4 = usr 5 = backup 6 = stand 8 = home Sun StorEdge™ Volume Manager array tags: 14 = public (region) 15 = private (region)
Flag	00 wm = The disk slice is writable and mountable. 01 wu = The disk slice is writable and unmountable. *This is the default state of slices dedicated for swap areas.* 10 rm = The disk slice is read-only and mountable. 11 ru = The disk slice is read-only and unmountable.
Cylinders	The starting and ending cylinder number for the disk slice.
Size	The slice size: Mbytes (MB), Gbytes (GB), blocks (b), or cylinders (c).
Blocks	The total number of cylinders and the total number of sectors per slice.

Partitioning a Disk

Caution – Do not change the size of disk slices that are currently in use.

The following steps demonstrate how to divide a disk into slices:

1. As the `root` user, type `format` at the prompt, and press Return.

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <ST34321A cyl 8892 alt 2 hd 15 sec 63>
          /pci@1f,0/pci@1,1/ide@3/dad@0,0
       1. c1t0d0 <SUN1.3G cyl 1965 alt 2 hd 17 sec 80>
          /pci@1f,0/pci@1/scsi@1/sd@0,0
Specify disk (enter its number):
```

   The `format` utility searches for all attached disks that are powered on. For each disk it finds, the `format` utility displays the logical device name, Sun marketing name, physical parameters, and physical device name.

2. Choose the second disk by selecting the number located to the left of that disk's logical device name. From the preceding display, the number chosen is 1. The `format` utility's main menu appears.

```
Specify disk (enter its number): 1
selecting c1t0d0
[disk formatted]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
```

```
        !<cmd>      - execute <cmd>, then return
        quit
format>
```

The specific menu selections that you can use to view, change, or commit disk slices include the following:

`partition`	Displays the Partition menu
`label`	Writes the current partition definition to the disk label
`verify`	Reads and displays the disk label
`quit`	Exits the `format` utility

3. Type `partition` at the `format` prompt. The Partition menu appears.

```
format> partition

PARTITION MENU:
        0      - change '0' partition
        1      - change '1' partition
        2      - change '2' partition
        3      - change '3' partition
        4      - change '4' partition
        5      - change '5' partition
        6      - change '6' partition
        7      - change '7' partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        !<cmd> - execute <cmd>, then return
        quit
```

The Partition menu enables you to perform the following functions:

`0-7`	Specify the offset and size of up to eight slices
`select`	Choose a predefined partition table from the `/etc/format.dat` file
`modify`	Change the current partition table in memory
`name`	Provide a means to identify the partition table in the `/etc/format.dat` file
`print`	Display the current partition table in memory
`label`	Write the current partition table to the disk label
`!<cmd>`	Escape from the utility and execute a command from the shell

4. Type `print` at the `partition` prompt to display the disk label that was copied to random access memory (RAM) when the `format` utility was invoked.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 1965 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0 unassigned    wm       0               0         (0/0/0)           0
  1 unassigned    wm       0               0         (0/0/0)           0
  2     backup    wu       0 - 1964        1.27GB    (1965/0/0) 2672400
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6 unassigned    wu       0               0         (0/0/0)           0
  7 unassigned    wm       0               0         (0/0/0)           0
```

The name of the partition table appears in parentheses in the first line of the table.

The columns of the table have the following meanings:

`Part`	The disk slice number
`Tag`	The predefined, optional tag
`Flag`	The predefined, optional flag
`Cylinders`	The starting and ending cylinder number for the slice
`Size`	The slice size in blocks (`b`), cylinders (`c`), Mbytes (`MB`), or Gbytes (`GB`)
`Blocks`	The total number of cylinders and the total number of sectors per slice

5. Select Slice 0 (zero) by entering `0`.

```
partition> 0
Part      Tag    Flag     Cylinders        Size            Blocks
  0 unassigned    wm       0               0          (0/0/0)           0
```

6. When prompted for the ID tag, type a question mark (`?`), and press Return to list the available choices. You can change a tag by entering a new tag name.

```
Enter partition id tag[unassigned]: ?
Expecting one of the following: (abbreviations ok):
        unassigned    boot          root          swap
        usr           backup        stand         var
        home          alternates

Enter partition id tag[unassigned]:
```

7. Type the tag `alternates`, and press Return.

```
Enter partition id tag[unassigned]: alternates
```

8. When prompted for the permission flags, type a question mark (`?`), and press Return to list the available choices. You can change a flag by entering the new flag name.

```
Enter partition permission flags[wm]: ?
Expecting one of the following: (abbreviations ok):
        wm    - read-write, mountable
        wu    - read-write, unmountable
        rm    - read-only, mountable
        ru    - read-only, unmountable

Enter partition permission flags[wm]:
```

9. Press Return to accept the default flag.

```
Enter partition permission flags[wm]: <return>
```

10. Press Return to accept the starting cylinder of 0 (zero).

```
Enter new starting cyl[0]: <return>
```

11. Enter 400mb for the new partition size for Slice 0.

```
Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: 400mb
```

12. Type print, and press Return. The Partition table appears.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 1965 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0 alternates    wm       0 -  602      400.43MB    (603/0/0)    820080
  1 unassigned    wm       0               0         (0/0/0)           0
  2     backup    wu       0 - 1964        1.27GB    (1965/0/0) 2672400
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6 unassigned    wu       0               0         (0/0/0)           0
  7 unassigned    wm       0               0         (0/0/0)           0
```

The current partition table shows the change to Slice 0.

Now adjust the starting cylinder for Slice 1.

13. Select slice number 1 by typing **1**.

```
partition>  1
Part      Tag    Flag     Cylinders        Size            Blocks
  1 unassigned    wm       0               0         (0/0/0)           0
```

14. Type the tag swap, and press Return.

```
Enter partition id tag[unassigned]: swap
```

15. Type wu at the permission flags selection, and press Return.

```
Enter partition permission flags[wm]: wu
```

16. Enter the new starting cylinder for Slice 1.

```
Enter new starting cyl[0]: 603
```

17. Enter the new partition size for Slice 1.

```
Enter partition size[0b, 0c, 603e, 0.00mb, 0.00gb]: 60mb
```

18. Type `print`, and press Return.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 1965 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0 alternates    wm       0 -  602      400.43MB    (603/0/0)    820080
  1       swap    wm     603 -  693       60.43MB    (91/0/0)     123760
  2     backup    wu       0 - 1964        1.27GB    (1965/0/0) 2672400
  3 unassigned    wm       0                0        (0/0/0)          0
  4 unassigned    wm       0                0        (0/0/0)          0
  5 unassigned    wm       0                0        (0/0/0)          0
  6 unassigned    wu       0                0        (0/0/0)          0
  7 unassigned    wm       0                0        (0/0/0)          0
```

The current partition table shows the change to Slice 1.

The new starting cylinder for Slice 1 is one greater than the ending cylinder for Slice 0.

Now adjust the starting cylinder for Slice 7.

19. Type **7** to select Slice 7.

```
partition> 7
Part      Tag    Flag     Cylinders        Size            Blocks
  7 unassigned    wm       0                0        (0/0/0) 0
```

20. Type the tag **home**, and press Return.

```
Enter partition id tag[unassigned]: home
```

21. Press Return to select the default flag.

```
Enter partition permission flags[wm]: <return>
```

22. Type the new starting cylinder for Slice 7.

```
Enter new starting cyl[0]: 694
```

23. Type the new partition size for Slice 7 by typing a dollar (`$`) sign.

```
Enter partition size[0b, 0c, 694e, 0.00mb, 0.00gb]: $
partition>
```

Note – Enter a dollar (`$`) sign as a value for the last partition size to automatically assign the remaining space on the disk to this slice.

---

24. Type **print** to display the partition table.

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 1965 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0 alternates    wm       0 -  602      400.43MB    (603/0/0)    820080
  1       swap    wm     603 -  693       60.43MB    (91/0/0)     123760
  2     backup    wu       0 - 1964        1.27GB    (1965/0/0) 2672400
  3 unassigned    wm       0               0         (0/0/0)          0
  4 unassigned    wm       0               0         (0/0/0)          0
  5 unassigned    wm       0               0         (0/0/0)          0
  6 unassigned    wu       0               0         (0/0/0)          0
  7       home    wm     694 - 1964      844.02MB    (1271/0/0) 1728560
```

Add up the cylinders in the Blocks column for Slice 0, Slice 1, and Slice 7. The number should equal the total number of cylinders contained in Slice 2.

25. After checking the partition table to ensure that there are no errors, label the disk by typing `label`.

```
partition> label
Ready to label disk, continue? y

partition>
```

Saving a Partition Table to the `/etc/format.dat` File

Use this optional procedure to add the newly created partition table to the `/etc/format.dat` file. You can save customized partition tables to the `/etc/format.dat` file and use them to quickly partition other disks of the same type on the system.

Note – Remember that, by default, the system saves the new partition information to the `./format.dat` file.You must enter the full path of the `/etc/format.dat` file to update the proper file.

To save a customized partition table, display the Partition menu and perform the following steps:

1. Type **name** to enter a unique name for the current partition table. Frequently, the disk manufacturer's name is used.

```
partition> name
Enter table name (remember quotes): SUN1.3G
```

Note – Quotes are only required if the partition table name is more than one word, for example, "SUN1.3G Generic."

2. Exit the Partition menu.

```
partition> quit
```

3. Type **save** to save the new partition table information, and enter the full path name for the /etc/format.dat file.

```
format> save
Saving new partition definition
Enter file name["./format.dat"]: /etc/format.dat
```

Using the Customized Partition Table

To retrieve a customized partition table, display the format menu and perform the following steps:

1. Type **partition**.

```
format> partition
```

2. Type **select** to display a list of customized partition tables, and choose the desired table by entering its assigned number.

```
partition> select
        0. SUN1.3G
        1. original
        2. SUN4.2
Specify table (enter its number)[3]: 0
```

3. Label the disk with the selected partition table.

```
partition> label
Ready to label disk, continue? yes
```

4. Exit the Partition menu.

```
partition> quit
```

5. Read the new disk label.

```
format> verify
Primary label contents:

Volume name = <        >
ascii name  = <SUN1.3G cyl 1965 alt 2 hd 17 sec 80>
pcyl        = 3500
ncyl        = 1965
acyl        =    2
nhead       =   17
nsect       =   80
Part      Tag    Flag     Cylinders        Size            Blocks
  0 alternates    wm       0 -  602      400.43MB    (603/0/0)   820080
  1       swap    wm     603 -  693       60.43MB    (91/0/0)    123760
  2     backup    wu       0 - 1964        1.27GB    (1965/0/0) 2672400
  3 unassigned    wm       0                0        (0/0/0)          0
  4 unassigned    wm       0                0        (0/0/0)          0
  5 unassigned    wm       0                0        (0/0/0)          0
  6 unassigned    wu       0                0        (0/0/0)          0
  7       home    wm     694 - 1964      844.02MB    (1271/0/0) 1728560
```

6. Exit the `format` utility.

```
format> quit
```

Managing Disk Labels

Every disk in the Solaris OE has a label set aside for storing information about the disk's controller, geometry, and slices.

Viewing the Disk VTOC

You can use two methods for locating and viewing a disk's label or VTOC:

- Use the `verify` command from the `format` utility
- Invoke the `prtvtoc` command from the command line

Reading a Disk's VTOC Using the `verify` Command

The `verify` command enables you to view a disk's VTOC from within the `format` utility. To read a disk's VTOC, perform the following steps:

1. At the `format` prompt, enter the `verify` command, and press Return.

```
format> verify

Primary label contents:

Volume name = <        >
ascii name  = <SUN1.3G cyl 1965 alt 2 hd 17 sec 80>
pcyl        = 3500
ncyl        = 1965
acyl        =    2
nhead       =   17
nsect       =   80
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm       0 -  602       400.43MB    (603/0/0)    820080
  1       swap    wm     603 -  693        60.43MB    (91/0/0)     123760
  2     backup    wm       0 - 1964         1.27GB    (1965/0/0) 2672400
  3 unassigned    wm       0                0         (0/0/0)           0
  4 unassigned    wm       0                0         (0/0/0)           0
  5 unassigned    wm       0                0         (0/0/0)           0
  6 unassigned    wm       0                0         (0/0/0)           0
  7       home    wm     694 - 1964       844.02MB    (1271/0/0) 1728560
```

2. Type `quit` or `q`, and press Return to exit the `format` menu.

Reading a Disk's VTOC Using the `prtvtoc` Command

The `prtvtoc` command enables you to view a disk's VTOC from the command line. To view a disk's VTOC from the command line, type the following:

```
# prtvtoc /dev/rdsk/c1t0d0s2
* /dev/rdsk/c1t0d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      80 sectors/track
*      17 tracks/cylinder
*    1360 sectors/cylinder
*    3500 cylinders
*    1965 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00          0     820080    820079
       1      3    00     820080     123760    943839
       2      5    00          0    2672400   2672399
       7      8    00     943840    1728560   2672399
```

The disk label information includes the following fields:

`Dimensions`	Describes the logical dimensions of the disk.
`Flags`	Describes the flags that are listed in the partition table.
`Partition`	A slice number. It is described further in Table 2-2 on page 2-38.
`Tag`	A value used to indicate how the slice is being used. It is described further in Table 2-2 on page 2-38.
`Flags`	The `00` flag is read/write, mountable; `01` is read/write, unmountable; and `10` is read only. These are described further in Table 2-2 on page 2-38.
`First Sector`	Defines the first sector of the slice.

`Sector Count`	Defines the total number of sectors in the slice.
`Last Sector`	Defines the last sector number in the slice.
`Mount Directory`	If the field is empty, the slice is currently not mounted and no entry exists in the `/etc/vfstab` file.

Relabeling a Disk

Save a disk's VTOC to a file by using the `prtvtoc` command. This allows you to relabel the disk by using the `fmthard` command if one of the following situations occurs:

- The VTOC on the disk has been destroyed.
- You accidentally changed the partition information on the disk and did not save a backup label in the `/etc/format.dat` file.

To save a disk's VTOC to a file, perform the command:

```
# prtvtoc /dev/rdsk/c1t0d0s2 > /vtoc/c1t0d0
```

The `fmthard` Command

To relabel a disk, you can save the output of the `prtvtoc` command into a file on another disk and use it as the `datafile` argument to the `fmthard` command.

```
fmthard -s datafile /dev/rdsk/c#t#d#s2
```

Caution – The `fmthard` command cannot write a disk label on an unlabeled disk. Use the `format` utility for this purpose.

If the need to relabel a disk arises and the VTOC was previously saved to a file, the following options are available:

- Run `format`, select the disk, and label it with the default partition table.
- Use the `fmthard` command to write the desired label information, previously saved to a *datafile* back to the disk.

```
# fmthard -s /vtoc/c1t0d0 /dev/rdsk/c1t0d0s2
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab is more difficult. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Working With Disks and Partitions (Level 1)

In this exercise, you complete the following tasks:

- Use the `format` utility to partition a disk
- Use the `prtvtoc` and `fmthard` commands to repair a corrupted disk label

Preparation

This exercise requires a system configured with an external disk.

Tasks

Complete the following tasks:

- Use the `format` command to list the disks currently attached to your system. Use the `prtvtoc` command to identify a disk that does not currently hold any mounted file systems. Examine the information that the `prtvtoc` command displays. Record the name of a disk that has no mount directory listed.

  (Steps 1–4 in the Level 2 lab)
- Use the `format` command to divide the unused disk into four slices of equal size. Use Slices 0, 1, 3, and 4. Set all other slices to size 0. Manually change the size of Slice 0 so that it ends 25 Mbytes into the space assigned to Slice 1.

  (Steps 4–11 in the Level 2 lab)
- Attempt to correct the overlap by using the Modify menu. Record the message that appears. Then correct the overlap by using the `all free hog` option. Verify your disk label with the `prtvtoc` command.

  (Steps 12–18 in the Level 2 lab)

- Create a directory called `/vtoc`. Run the `prtvtoc` command to read the label of the disk you modified, and save its output in a file in the `/vtoc` directory. Use the `dd` command from Step 21 of the Level 2 lab to destroy the label on the same disk. Attempt to read the disk label by using the `prtvtoc` command, and record the result. If required, use the `format` command to write a default label to the disk. Use the `fmthard` command to restore the label by using the output from the `prtvtoc` command that you saved earlier. Verify that the new label exists.

  (Steps 19–25 in the Level 2 lab)

Exercise: Working With Disks and Partitions (Level 2)

In this exercise, you complete the following tasks:

- Use the `format` utility to partition a disk
- Use the `prtvtoc` and `fmthard` commands to repair a corrupted disk label

Preparation

This exercise requires a system configured with an external disk.

Task Summary

In this exercise, you accomplish the following:

- Use the `format` command to list the disks currently attached to your system. Use the `prtvtoc` command to identify a disk that does not currently hold any mounted file systems. Examine the `Mount Directory` field in the information that the `prtvtoc` command displays. Record the name of a disk that has no mount directory listed.
- Use the `format` command to divide the unused disk into four slices of equal size. Use Slices 0, 1, 3, and 4. Set all other slices to size 0. Manually change the size of Slice 0 so that it ends 25 Mbytes into the space assigned to Slice 1.
- Attempt to correct the overlap by using Option 0 from the Modify menu. Record the message that appears. Then correct the overlap by using the `all free hog` option. Verify your disk label with the `prtvtoc` command.
- Create a directory called `/vtoc`. Run the `prtvtoc` command to read the label of the disk you modified, and save its output in a file in the `/vtoc` directory. Use the `dd` command to destroy the label on the same disk. Attempt to read the disk label by using the `prtvtoc` command, and record the result. If required, use the `format` command to write a default label to the disk. Use the `fmthard` command to restore the label by using the output from the `prtvtoc` command that you saved earlier. Verify that the new label exists.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Run the `format` command.
2. Record the list of disks presented by the `format` command, for example, `c0t0d0` and `c1t0d0`.

   Press the Control-D keys to exit the `format` utility.
3. Use the `prtvtoc` command to list the VTOC for each of the disks that you found in the previous step. Examine the `Mount Directory` field in the information that the `prtvtoc` command displays. Record the name of a disk that has no mount directory listed. This is an unused disk.
4. Run the `format` command again. Select the unused disk from the list of disks presented.
5. Display the Partition menu. Print the current partition table, and record the number of megabytes assigned to Slice 2. For example, if the disk reports 4 Gbytes, record 4000 Mbytes.

   Mbytes:
6. Divide the number of megabytes by 4. Use the result as the number of megabytes to assign as disk space to four slices. Round down to the next whole megabyte if the result includes a fraction.

   Mbytes/4:
7. Display the Partition menu again. Select Slice 0. Accept the defaults for tags and flags. Start this first slice on Cylinder 0. Enter the resulting number of megabytes from the previous step for the slice size. Print the partition table again to verify the change.
8. Set the sizes of Slices 1, 3, and 4 so that they are the same as Slice 0. Begin each successive slice on the cylinder that follows the ending cylinder of the previous slice.
9. Set Slices 5, 6, and 7 to start at Cylinder 0, and assign them 0 Mbytes.
10. Print the partition table. Is there any overlap of ending and beginning cylinders for any of the slices listed? Proceed to the following steps to introduce this problem.

11. Add 25 to the number Mbytes/4 value listed in Step 6.

    (Mbytes/4) + 25:

    Change Slice 0 so that it uses the new size listed above.

    The partition table should now indicate that Slice 0 ends after Slice 1 begins.

12. Use the `modify` command from the Partition menu to attempt to fix this problem. Select Item `0` to modify the current partition table.

    Which warnings appear?

13. Modify the partition table. Select Item `1` to use the `All Free Hog` method.

14. The partition table appears. Observe the Cylinders and Size columns, and notice that they are all zero.

15. Respond to the prompts to continue the process. Select Slice 4 as the `All Free Hog` slice. Use the size listed in Step 6 for Slices 0, 1, and 3. Set the other slices to Size 0.

    At the end of this process, you should have three slices of equal size, where Slice 4 takes up any extra room if it exists.

16. Name the partition table `"SA239partition"`, then label the disk.

17. Quit the partition menu, and save your new partition table to the `/etc/format.dat` file. Carefully read the message that is displayed by the `format` utility, and enter the correct file name. Quit the `format` utility when you have finished. Use the `cat` command to view the contents of the `/etc/format.dat` file. Note that your information is appended to the file.

18. Verify your new partition table with the `prtvtoc` command.

19. Create a directory called `/vtoc`.

20. Use the `prtvtoc` command to print the partition table that you just created, and save its output to a file in the `/vtoc` directory. Name the file so that it corresponds with the disk you are examining. Use the `cat` command to verify that valid information exists in the file that you create.

21. Use the following dd command to destroy the disk label. Be certain to specify the correct disk device name for the of= argument. Enter all other arguments exactly as listed.

```
# dd if=/dev/zero of=/dev/rdsk/c1t0d0s2 bs=512 count=1
1+0 records in
1+0 records out
#
```

22. Attempt to read the label from the same disk by using the prtvtoc command.

    What happens?

23. If the prtvtoc command reported an "Unable to read Disk geometry" message, use the format command to place a default label on the disk for which you destroyed the label earlier.

    If the prtvtoc command reports that only Slice 2 exists on the disk, skip to the next step. Otherwise, perform the commands:

```
# format
Searching for disks...done

c1t0d0: configured with capacity of 4.00GB


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <Seagate Medalist 34342A cyl 8892 alt 2 hd 15 sec 63>
          /pci@1f,0/pci@1,1/ide@3/dad@0,0
       1. c1t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
          /pci@1f,0/pci@1/pci@2/SUNW,isptwo@4/sd@3,0
Specify disk (enter its number): 1
selecting c1t0d0
[disk formatted]
Disk not labeled.  Label it now? Y

(format menu)

format> q
#
# prtvtoc /dev/rdsk/c1t0d0s2
```

24. Use the fmthard command to write to the disk the label information you saved earlier.

25. Attempt to read the label from the same disk.

    Was this successful?

Exercise: Working With Disks and Partitions (Level 3)

In this exercise, you complete the following tasks:

- Use the `format` utility to partition a disk
- Us the `prtvtoc` and `fmthard` commands to repair a corrupted disk label

Preparation

This exercise requires a system configured with an external disk.

Task Summary

In this exercise, you accomplish the following:

- Use the `format` command to list the disks currently attached to your system. Use the `prtvtoc` command to identify a disk that does not currently hold any mounted file systems. Examine the `Mount Directory` field in the information that the `prtvtoc` command displays. Record the name of a disk that has no mount directory listed.
- Use the `format` command to divide the unused disk into four slices of equal size. Use Slices 0, 1, 3, and 4. Set all other slices to size 0. Manually change the size of Slice 0 so that it ends 25 Mbytes into the space assigned to Slice 1.
- Attempt to correct the overlap using Option 0 from the Modify menu. Record the message that appears. Then correct the overlap by using the `all free hog` partition. Verify your disk label with the `prtvtoc` command.
- Create a directory called `/vtoc`. Run the `prtvtoc` command to read the label of the disk you modified, and save its output in a file in the `/vtoc` directory. Use the `dd` command to destroy the label on the same disk. Attempt to read the disk label by using the `prtvtoc` command, and record the result. If required, use the `format` command to write a default label to the disk. Use the `fmthard` command to restore the label by using the output from the `prtvtoc` command that you saved earlier. Verify that the new label exists.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Run the `format` command.

```
# format
```

2. Record the list of disks presented by the `format` command, for example, `c0t0d0` and `c1t0d0`.

   Press Control-D to exit the `format` utility.

```
format> Control-D
#
```

3. Use the `prtvtoc` command to list the VTOC for each of the disks you found in the previous step. Examine the `Mount Directory` field in the information that the `prtvtoc` command displays. Record the name of a disk that has no mount directory listed. This will be an unused disk.

```
# prtvtoc /dev/rdsk/c1t0d0s2
```

   Unused disk: *Your entry will depend on your system.*

4. Run the `format` command again. Select the unused disk from the list of disks presented.

```
# format
(list of disks)
Specify disk (enter its number): x
```

5. Display the Partition menu. Print the current partition table, and record the number of megabytes assigned to Slice 2. For example, if the disk reports 4 Gbytes, record 4000 Mbytes.

```
format> part
partition> print
```

   Mbytes: *Your entry will depend on your system.*

6. Divide the number of megabytes by 4. Use the result as the number of megabytes to assign as disk space to four slices. Round down to the next whole megabyte if the result includes a fraction.

   Mbytes/4: *Your entry will depend on your system.*

7. Display the Partition menu again. Select Slice 0. Accept the defaults for tags and flags. Start this first slice on Cylinder 0. Enter the resulting number of megabytes from the previous step for the slice size. Print the partition table again to verify the change.

```
partition> 0
Part      Tag    Flag     Cylinders        Size            Blocks
  0 unassigned    wm       0               0         (0/0/0)          0

Enter partition id tag[unassigned]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 0
Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: 300mb
partition> print
(partition table)
```

8. Set the sizes of Slices 1, 3, and 4 so that they are the same as Slice 0. Begin each successive slice on the cylinder that follows the ending cylinder of the previous slice.

```
partition> ?
(Partition menu)
partition> 1
Part      Tag    Flag     Cylinders        Size            Blocks
  1 unassigned    wm       0               0         (0/0/0)          0
Enter partition id tag[unassigned]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 452
Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: 300mb
partition> print
(partition table)
```

9. Set Slices 5, 6, and 7 to start at Cylinder 0, and assign them 0 Mbytes.

```
partition> ?
(Partition menu)
partition> 5
Part      Tag    Flag     Cylinders        Size            Blocks
  5 unassigned    wm       0               0         (0/0/0)          0

Enter partition id tag[unassigned]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 0
Enter partition size[0b, 0c, 0e, 0.00mb, 0.00gb]: 0m
partition>
```

10. Print the partition table. Is there any overlap of ending and beginning cylinders for any of the slices listed? Proceed to the following steps to introduce this problem.

```
partition> print
```

11. Add 25 to the number Mbytes/4 value listed in Step 6.

    (Mbytes/4) + 25: *Your entry will depend on your system.*

    Change Slice 0 so that it uses the new size listed previously.

```
partition> ?
(Partition menu)
partition> 0
Part      Tag    Flag     Cylinders        Size            Blocks
0 unassigned    wm       0 -  451      300.16MB    (452/0/0)   614720

Enter partition id tag[unassigned]: <Return>
Enter partition permission flags[wm]: <Return>
Enter new starting cyl[0]: 0
Enter partition size[614720b, 452c, 451e, 300.16mb, 0.29gb]: 325mb
partition> print
(partition table)
```

The partition table should now indicate that Slice 0 ends after Slice 1 begins.

12. Use the `modify` command from the Partition menu to attempt to fix this problem. Select Item `0` to modify the current partition table.

```
partition> ?
(Partition menu)
partition> modify
Select partitioning base:
       0. Current partition table (unnamed)
       1. All Free Hog
Choose base (enter number) [0]? 0
```

Which warnings display?

```
Warning: Overlapping partition (1) in table.
Warning: Fix, or select a different partition table.
```

13. Modify the partition table. Select Item `1` to use the `All Free Hog` option.

```
partition> modify
Select partitioning base:
        0. Current partition table (original)
        1. All Free Hog
Choose base (enter number) [0]? 1
```

14. The partition table appears. Observe the Cylinders and Size columns, and notice that they are all zero; for example:

```
Part      Tag    Flag     Cylinders         Size            Blocks
  0       root    wm      0                 0          (0/0/0)            0
  1       swap    wu      0                 0          (0/0/0)            0
  2     backup    wu      0 - 1964          1.27GB     (1965/0/0) 2672400
  3 unassigned    wm      0                 0          (0/0/0)            0
  4 unassigned    wm      0                 0          (0/0/0)            0
  5 unassigned    wm      0                 0          (0/0/0)            0
  6        usr    wm      0                 0          (0/0/0)            0
  7 unassigned    wm      0                 0          (0/0/0)            0
```

15. Respond to the prompts to continue the process. Select Slice 4 as the `All Free Hog` partition. Use the size listed in Step 6 for Slices 0, 1, and 3. Set the other slices to Size 0.

```
Do you wish to continue creating a new partition
table based on above table[yes]? yes
Free Hog partition[6]? 4
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 300mb
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 300mb
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]: 300mb
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]: 0
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]: 0
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]: 0
```

At the end of this process, you should have three slices of equal size, where Slice 4 takes up any extra room if it exists.

16. Name the partition table `"SA239partition"`, then label the disk.

```
Okay to make this the current partition table[yes]? y
Enter table name (remember quotes): "SA239partition"

Ready to label disk, continue? y

partition>
partition> quit
(format menu)
format>
```

17. Save your new partition table to the /etc/format.dat file. Carefully read the message that is displayed by the format utility, and enter the correct file name. Quit the format utility when you have finished. Use the cat command to view the contents of the /etc/format.dat file. Note that your information is appended to the file.

```
format> save
Saving new disk and partition definitions
Enter file name["./format.dat"]: /etc/format.dat
format> quit
#
# cat /etc/format.dat
```

18. Verify your new partition table with the prtvtoc command.

```
# prtvtoc /dev/rdsk/c1t0d0s2
```

19. Create a directory called /vtoc.

```
# mkdir /vtoc
```

20. Use the prtvtoc command to print the partition table that you just created, and save its output to a file in the /vtoc directory. Name the file so that it corresponds with the disk you are examining. Use the cat command to verify that valid information exists in the file that you create.

```
# prtvtoc /dev/rdsk/c1t0d0s2 > /vtoc/c1t0d0
# cat /vtoc/c1t0d0
```

21. Use the following dd command to destroy the disk label. Be certain to specify the correct disk device name for the of= argument. Enter all other arguments exactly as listed.

```
# dd if=/dev/zero of=/dev/rdsk/c1t0d0s2 bs=512 count=1
1+0 records in
1+0 records out
#
```

22. Attempt to read the label from the same disk by using the prtvtoc command.

```
# prtvtoc /dev/rdsk/c1t0d0s2
```

What happens?

Different disk types present different results. SCSI disks might report messages that indicate that the disk label is unreadable, for example:

```
prtvtoc: /dev/rdsk/c1t0d0s2: Unable to read Disk
geometry
```

IDE disks might report a partition table where only Slice 2 remains defined, for example:

```
Partition  Tag  Flags     Sector      Count                  Sector
  Mount Directory
          2                     5     01                          0
       17801280  17801279
```

23. If the prtvtoc command reported an "Unable to read Disk geometry" message, use the format command to place a default label on the disk for which you destroyed the label earlier.

    If the prtvtoc command reports that only Slice 2 exists on the disk, skip to the next step. Otherwise, perform the commands:

```
# format
Searching for disks...done

c1t0d0: configured with capacity of 1.3GB


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <Seagate Medalist 34342A cyl 8892 alt 2 hd 15 sec 63>
          /pci@1f,0/pci@1,1/ide@3/dad@0,0
       1. c1t0d0 <SUN1.3G cyl 1965 alt 2 hd 17 sec 80>
          /pci@1f,0/pci@1/scsi@1/sd@0,0
Specify disk (enter its number): 1
selecting c1t0d0
[disk formatted]
Disk not labeled.  Label it now? Y

(format menu)
format> q
#
# prtvtoc /dev/rdsk/c1t0d0s2
```

24. Use the fmthard command to write to the disk the label information you saved earlier.

```
# fmthard -s /vtoc/c1t0d0 /dev/rdsk/c1t0d0s2
fmthard:  New volume table of contents now in place.
#
```

25. Attempt to read the label from the same disk.

```
# prtvtoc /dev/rdsk/c1t0d0s2
```

Was this successful?

This command should successfully read the disk label.

Introducing the Solaris™ Management Console

The Solaris Management Console is a Java technology-based tool for the administration of systems. It provides a central integration point for the configuration and administration of important applications and services.

The Solaris Management Console software simplifies the job of configuring and administering servers. With point-and-click graphical user interface (GUI) tools, the Solaris Management Console makes the Solaris OE easy to administer, especially for administrators who are not familiar with the UNIX environment.

Starting the Solaris Management Console

The Solaris Management Console can be started from the command line or from within the Application Manager by clicking the Solaris Management Console icon.

Log in to your system as `root`, and type **`smc&`** in a terminal window. You can start the Solaris Management Console as a normal user, but some tools and applications are not available to you. When you initiate the Solaris Management Console for the first time, it can take a few minutes to launch.

Note – The information provided in this course is only a small subset of the overall capabilities of the Solaris Management Console.

Using the Solaris Management Console Tools

The default toolbox for a Solaris Management Console server includes the following folders and tools:

System Status	This category includes System Information, Log Viewer, Processes, and Performance.
System Configuration	This category includes Users, Projects, Computers and Networks, and Patches.
Services	This category includes Scheduled Jobs.
Storage	This category includes Mounts and Shares, Disks, and Enhanced Storage.
Devices and Hardware	This category includes Serial Ports.

The Solaris Management Console enables local users and administrators to register remote Solaris Management Console servers and applications on the network they want to administer. When you access the Solaris Management Console, it dynamically configures tree views of those registered hosts and services. Point and click with the mouse to invoke an application remotely on a selected Solaris Management Console server and view the application's GUI on the local display.

Introducing the Help Screen

The online help for the Solaris Management Console provides an alternative to standard documentation. The information panes that appear in both the Solaris Management Console and the Solaris Management Console Toolbox Editor provide the steps necessary to perform the tasks executed within these windows. In addition, the Help menu item Contents displays a window that further describes the features and functions of the window components.

Figure 2-14 shows the help functionality of the Solaris Management Console.

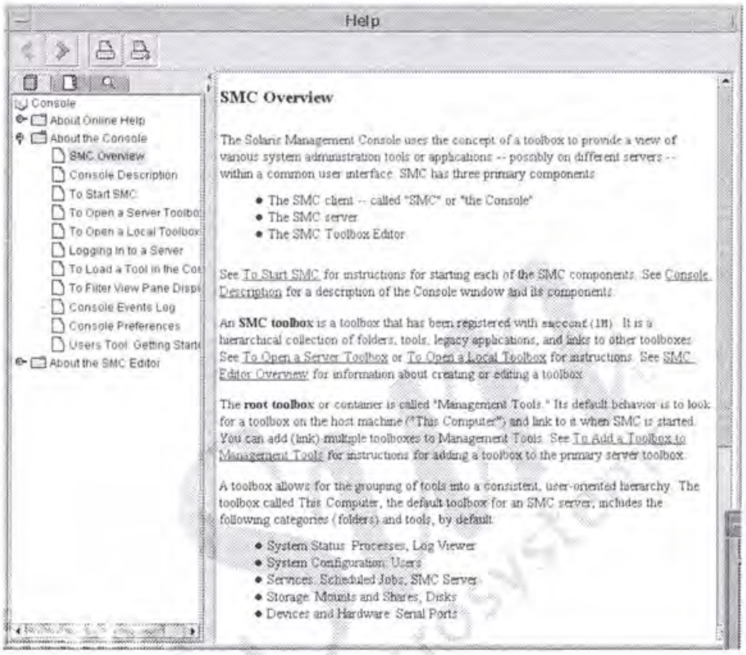

Figure 2-14 Solaris Management Console Help View

Restarting the Solaris Management Console

If you have trouble accessing Solaris Management Console, the reason might be that the Solaris Management Console server is not running or is in a problem state.

To determine if the Solaris Management Console server is running, as the `root` user, perform the command:

```
# /etc/init.d/init.wbem status
```

If the Solaris Management Console server is running, a response similar to the following returns: "`Solaris Management Console server version 2.1.0 running on port 898.`"

To stop the Solaris Management Console server, as the `root` user, perform the command:

```
# /etc/init.d/init.wbem stop
```

The following response returns: "`SMC stopped.`"

To start the Solaris Management Console server, as the `root` user, perform the command:

```
# /etc/init.d/init.wbem start
```

After a short time, the following response returns: "`SMC server started.`"

Identifying the Functional Areas of the Solaris Management Console

The Solaris Management Console and the Solaris Management Console Toolbox Editor windows are divided into functional areas as follows:

- Navigation pane
- View pane
- Information pane
- Location bar
- Status bar

Figure 2-15 shows these divisions.

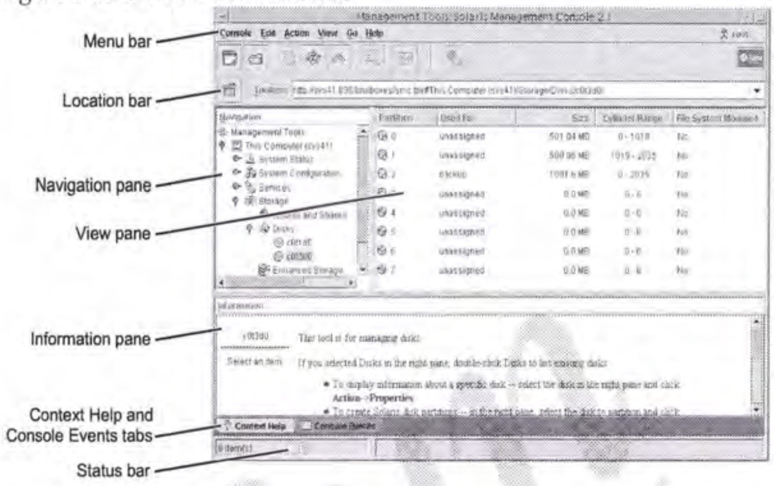

Figure 2-15 Solaris Management Console Overview

Note – The Location bar does not appear by default when you first launch the Solaris Management Console. Click View on the Menu bar, select the Show option, and select the Location option to display the Location bar.

Navigation Pane

The Navigation pane works like a frame in a web page. Clicking an item in the Navigation pane determines what appears in the View pane. The turner icon is displayed to the left of items that represent a group of items. Click the icon or the item to expand or collapse the group.

The Navigation pane is displayed or not displayed, depending on the Show setting in the View menu. Click View on the Menu bar, select the Show option, and select or deselect the Navigation option.

View Pane

The View pane displays the contents of the node selected in the Navigation pane. The contents could be a folder or a tool.

If the node selected in the Navigation pane is a folder, the View pane displays the contents of that folder.

If the node selected is a simple tool, such as the Process tool, the View pane displays a list of current processes. If the node selected is a complex tool, such as User Manager, the View pane displays additional tools, such as the tools for user accounts and email accounts. Select one of the additional tools, such as the user accounts node, and the View pane displays the contents of the tool.

Information Pane

The Information pane at the bottom of the Solaris Management Console window displays either context help for the object selected in the Navigation pane or a list of events and alarms for all Solaris Management Console events.

The Context Help tab and Console Events tab determine what is shown in the Information pane. Click the Context Help tab to display context help for the object selected. Click the Console Events tab to display a list of events and alarms for all Console events.

The Information pane is displayed or not displayed, depending on the Show setting in the View menu. Click View on the Menu bar, select the Show option, and select or deselect the Information option.

Location Bar

The Location bar, beneath the tool bar in the Solaris Management Console window, displays a Home Toolbox icon and a Toolbox field. Click the Home Toolbox icon to open the home toolbox. The Toolbox field indicates the current toolbox and the item currently selected in the toolbox. Click the button to the right of the Toolbox field to display a pull-down menu of recent toolboxes visited. Select a toolbox from the pull-down menu to open that toolbox.

The Location bar is displayed or not displayed, depending on the Show setting in the View menu. Click View on the Menu bar, select the Show option, and select or deselect the Location option.

Status Bar

The Status bar, located across the bottom of the Solaris Management Console window, displays three panes. The left pane of the Status bar indicates the number of nodes directly subordinate to the node selected in the Navigation pane. The center pane of the Status bar indicates Console activity. A moving bar inside the center pane functions as an activity indicator when Console activity occurs. The right pane of the Status bar provides progress information during some Console tasks.

The Status Bar is displayed or not displayed, depending on the Show setting in the View menu. Click View on the Menu bar, select the Show option, and select or deselect the Status bar option.

Partitioning a Disk by Using the Solaris Management Console Disks Manager Tool

The following section describes how to partition a disk by using the Solaris Management Console Disks Manager Tool (from this point on, referred to as the Disks Tool).

Partitioning the Disk Using the Disks Tool

To partition a disk by using the Disks Tool, you must first locate the Storage folder within the Navigation pane. The Storage folder consists of the Mounts and Shares folder, the Disks Tool, and the Enhanced Storage tools.

Use the Disks Tool to perform the following tasks:

- Display information about a specific disk
- Create Solaris OE disk partitions
- List partitions
- Copy the layout of one disk to another disk of the same type
- Change the disk's label

Perform the following steps to partition a disk by using Disks Tool:

1. Click Storage and then the Disks Tool. The Log In: User Name window appears, prompting you to enter the `root` password.

   Figure 2-16 shows the Log In: User Name window.

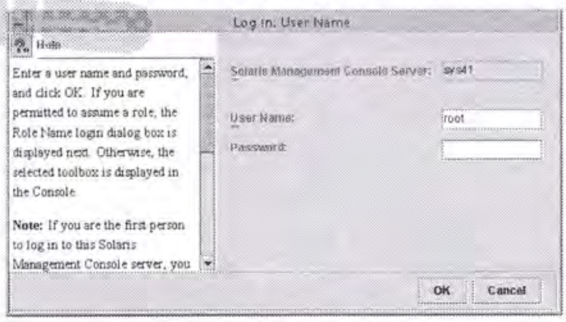

Figure 2-16 Log In: User Name Window

Figure 2-17 shows the Solaris Management Console after you have opened the Storage folder and then the Disks Tool. The figure shows a system with two disks.

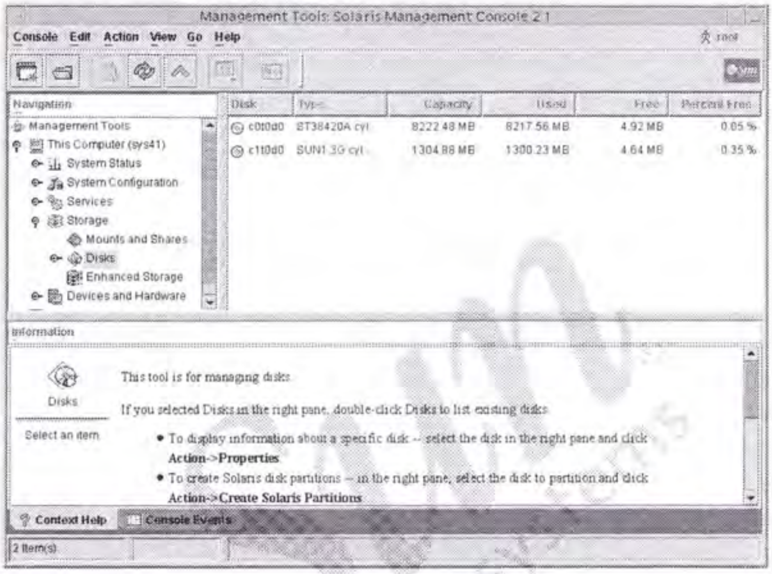

Figure 2-17 Management Tools: Solaris Management Console Window

2. Click to select a specific disk. Then click the Action menu on the Menu bar.

   The Action menu displays a list of functions that this window performs.

3. To display a graphical representation of a disk's partitioning, select the Properties option from the Action menu.

Figure 2-18 shows a 1.3 Gbyte drive.

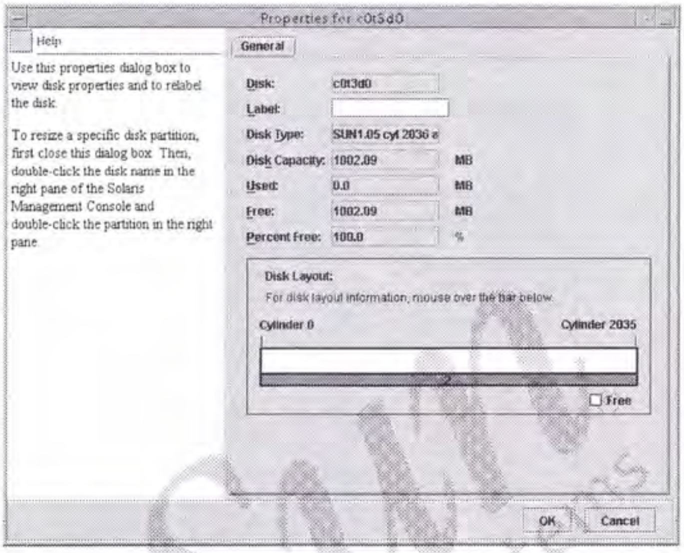

Figure 2-18 Properties Window

Basic disk information, including size, address, and available free space, is reported. Move the cursor over any partition on the Disk Layout bar to see the size and geometry of the partition slice in a pop-up window.

4. To create a new partition map on a disk, select the Create Solaris Partitions option from the Action menu.

Figure 2-19 shows the first window that you use to create partitions on a disk. This window prompts you to choose between creating custom-sized partitions and creating equal-sized partitions. In the figure, Create Custom-Sized Partitions is selected.

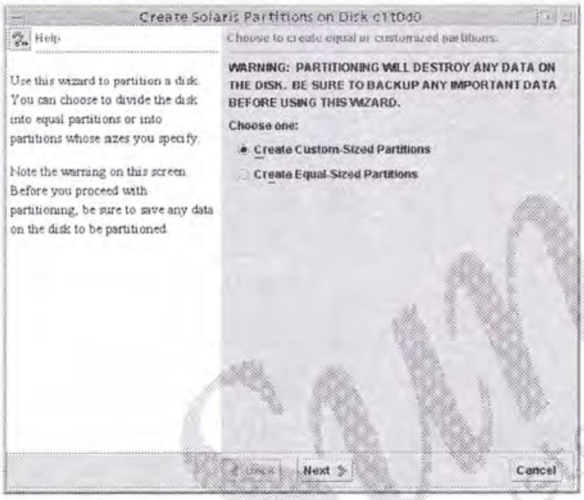

Figure 2-19 Create Solaris Partitions on Disk Window

5. Click Next after choosing how to divide the disk.

Figure 2-20 shows the next window that you use to create partitions on a disk. You are prompted to select the number of partitions. You can select up to seven partitions. You can also create some of them as zero-length partitions.

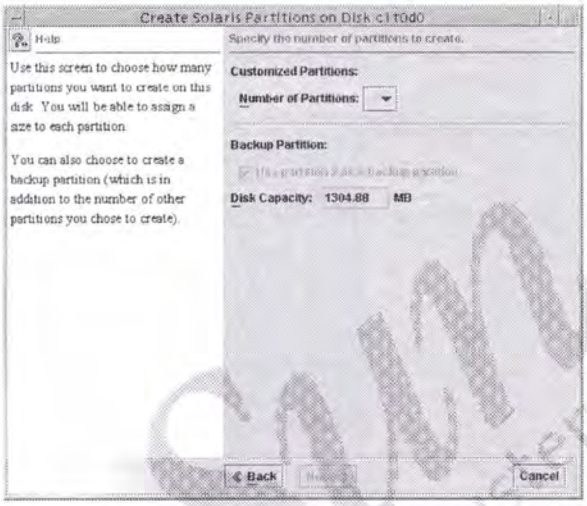

Figure 2-20 Create Solaris Partitions on Disk Window – Specify Number of Partitions

6. After selecting the number of partitions, click Next.

Figure 2-21 shows the window that enables you to display each partition. When a partition is displayed, the size of the partition is also displayed. You can choose to display the size of the partition in either a percentage of the disk space or the total number of megabytes, and you can adjust the size of each partition. The disk layout bar graphically represents the disk partitions. Place the cursor over the bar to view the amount of space that remains to be partitioned.

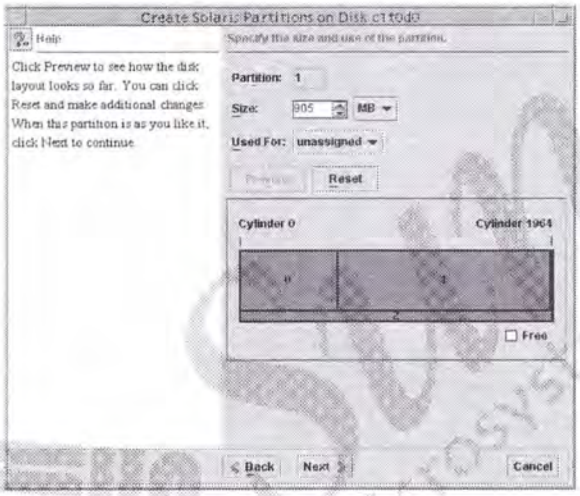

Figure 2-21 Create Solaris Partitions on Disk Window – Specify Size and Use of Partitions

7. Use this window to adjust the size of each partition to the desired size. Click Next when you have finished sizing the partitions.

Figure 2-22 shows the window that allows you to specify the partitions on which to create file systems.

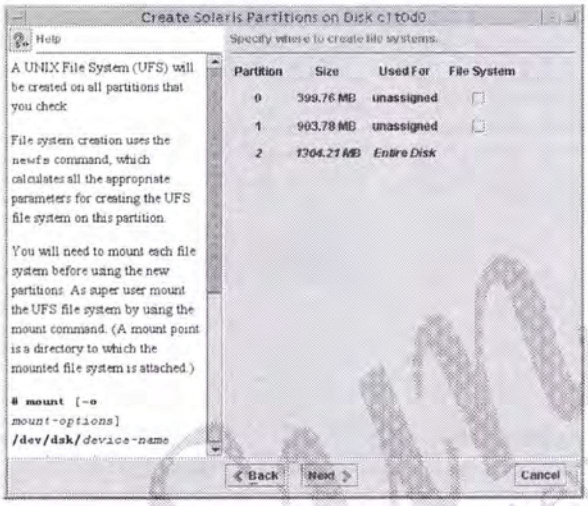

Figure 2-22 Create Solaris Partitions on Disk Window – Specify Where to Create File Systems

8. In the Create Solaris Partitions on Disk window, check the box under the file system that corresponds to each partition you want to use. Click Next when you are finished making your selections.

Figure 2-23 displays a list of the disk partitions you have created.

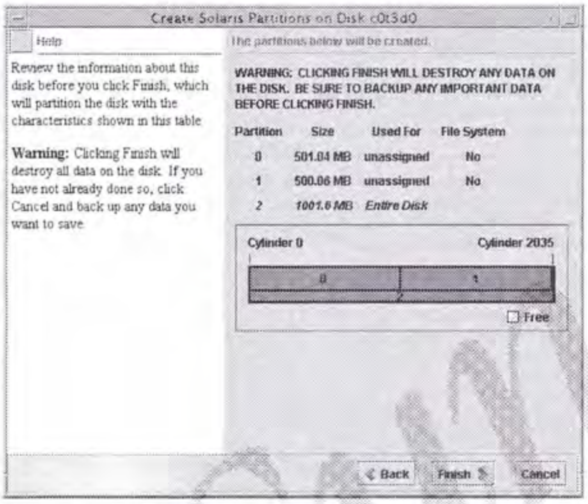

Figure 2-23 Create Solaris Partitions on Disk Window – Confirmation

9. If you are satisfied with the partitions, click Finish. The new partitioning is written, and the `newfs` utility runs on the partitions you selected to create a new file system.

Figure 2-24 displays the disks window of the Solaris Management Console after you have completed partitioning the disk. The created partitions are displayed in the Management Tools: Solaris Management Console window.

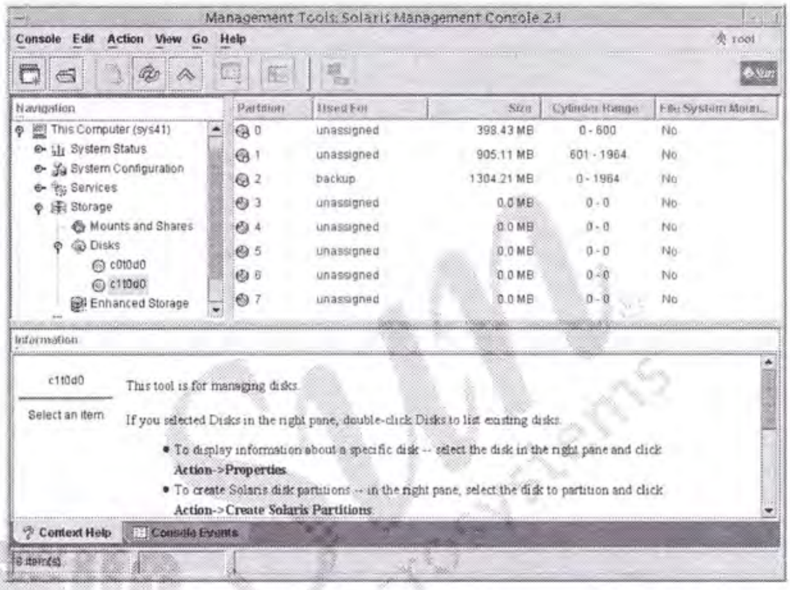

Figure 2-24 Management Tools: Solaris Management Console Window – Partitioning Completed

Performing the Exercises

You have the option to complete either of these labs. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab is guided. Although each step describes what you should do, you must determine which commands (and options) to input.

There are only two levels of this lab due to the nature of working within the Solaris Management Console GUI. Should you require assistance with any of the steps, consult the help functionality from the Solaris Management Console.

Exercise: Working With the Solaris Management Console (Level 1)

In this exercise, you complete the following tasks:

- Launch the Solaris Management Console Disks Manager Tool
- Partition the second drive of your system to match the boot drive

Preparation

This exercise requires a system with at least two disks, one of which is available for the student to re-partition.

Tasks

Complete the following tasks:

- Launch the Solaris Management Console, and choose the Disks Tool from the Storage folder.
- Authenticate as the `root` user by typing the `root` password.
- View information about your boot drive from the Disks Tool, and make note of it.
- On your spare hard drive, make four equal sized partitions on Slices 0, 1, 3, and 4.

Exercise: Working With the Solaris Management Console (Level 2)

In this exercise, you complete the following tasks:

- Launch the Solaris Management Console Disks Manager Tool
- Partition the second drive of your system to match the boot drive

Preparation

This exercise requires a system with at least two disks, one of which is available for the student to re-partition.

Task Summary

Complete the following tasks:

- Launch the Solaris Management Console, and choose the Disks Tool from the Storage folder.
- Authenticate as the `root` user by typing the `root` password.
- View information about your boot drive from the Disks Tool, and make note of it.
- On your spare hard drive, make four equal sized partitions on Slices 0, 1, 3, and 4.

Tasks

Complete the following steps:

1. Launch the Solaris Management Console from the command line or by using Application Manager.
2. Open the Disks Tool.
3. Select your boot drive from the Disks Tool, and record the partition information listed.
4. Select your spare drive from the Disks Tool. Select Create Solaris Partitions from the Action menu.
5. Choose Create Equal-Sized Partitions, and click Next.

6. Specify Number of Partitions as 4. Click Next.
7. Verify that you have four equal-sized partitions on Slices 0, 1, 3, and 4. Click Next.
8. Check the box beside Slice 4 to create a File System, and click Next.
9. After reviewing your choices and verifying that they are correct, click Finish.

   The Solaris Management Console window refreshes, and you should see the four equal-sized partitions listed in the View Pane.
10. Exit from the Solaris Management Console.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 3

Managing the Solaris OE File System

Objectives

Upon completion of this module, you should be able to:

- Describe Solaris OE file systems
- Create a new `ufs` file system
- Check the file system by using the `fsck` command
- Resolve file system inconsistencies
- Monitor file system use

The following course map shows how this module fits into the current instructional goal.

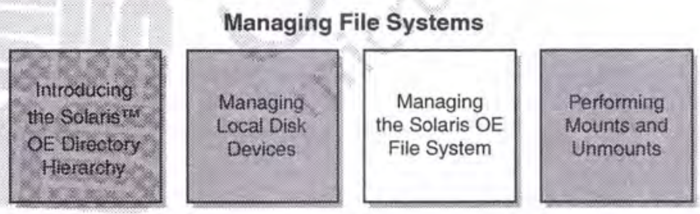

Figure 3-1 Course Map

Introducing Solaris OE File Systems

A file system is a collection of files and directories that make up a structured set of information. The Solaris OE supports three different types of file systems:

- Disk-based file systems
- Distributed file systems
- Pseudo file systems

Disk-based File Systems

Disk-based file systems are found on hard disks, CD-ROMs, diskettes, and DVDs. The following are examples of disk-based file systems:

- `ufs` – The UNIX file system in the Solaris OE is based on the Berkeley fast file system.
- hsfs – The High Sierra file system is a special-purpose file system developed for use on CD-ROM media.
- pcfs – The PC file system is a UNIX implementation of the disk operating system (DOS) file allocation table (FAT32) file system. The pcfs file system allows the Solaris OE to access PC-DOS formatted file systems. Users can use UNIX commands for direct read and write access to PC-DOS files.
- udfs – The Universal Disk Format file system is used for optical storage targeted at DVD and CD-ROM media. The UDF file system allows universal data exchange and supports read and write operations.

Distributed File Systems

Distributed file systems provide network access to file system resources.

- NFS – The network file system allows users to share files among many types of systems on the network. The NFS file system makes part of a file system on one system appear as though it were part of the local directory tree.

Pseudo File Systems

Pseudo file systems are memory based. These file systems provide for better system performance, in addition to providing access to `kernel` information and facilities. Pseudo file systems include:

- tmpfs – The temporary file system stores files in memory, which avoids the overhead of writing to a disk-based file system. The tmpfs file system is created and destroyed every time the system is rebooted.
- swapfs – The swap file system is used by the `kernel` to manage swap space on disks.
- fdfs – The file descriptor file system provides explicit names for opening files by using file descriptors (for example, `/dev/fd/0`, `/dev/fd/1`, `/dev/fd/2`) in the `/dev/fd` directory.
- procfs – The process file system contains a list of active processes in the `/proc` directory. The processes are listed by process number. Information in this directory is used by commands, such as the `ps` command.
- mntfs – The mount file system provides read-only information from the `kernel` about locally mounted file systems.

Creating a New ufs File System

This section describes the ufs file system in the Solaris OE.

Viewing the Solaris OE ufs File System

The user views the ufs file system differently than the operating system does in the Solaris OE. To a user, a file system appears as a collection of files and directories used to store and organize data for access by the system and its users. To the operating system, a file system is a collection of control structures and data blocks that occupy the space defined by a partition, which allow for data storage and management.

The Solaris OE stores data in a logical file hierarchy often consisting of several file systems. This file hierarchy is referred to as the Solaris directory hierarchy.

Figure 3-2 shows the Solaris OE hierarchy beginning with the / (root) directory.

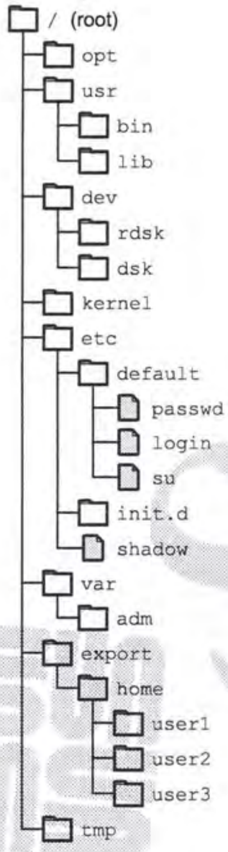

Figure 3-2 Solaris OE Directory Hierarchy

Note – Figure 3-2 is not a complete representation of a Solaris OE directory hierarchy.

A ufs file system is created on a disk slice before it is used in the Solaris OE. Creating a ufs file system on a disk slice enables the Solaris OE to store UNIX directories and files.

Figure 3-3 shows how the ufs file systems are located on various disk slices.

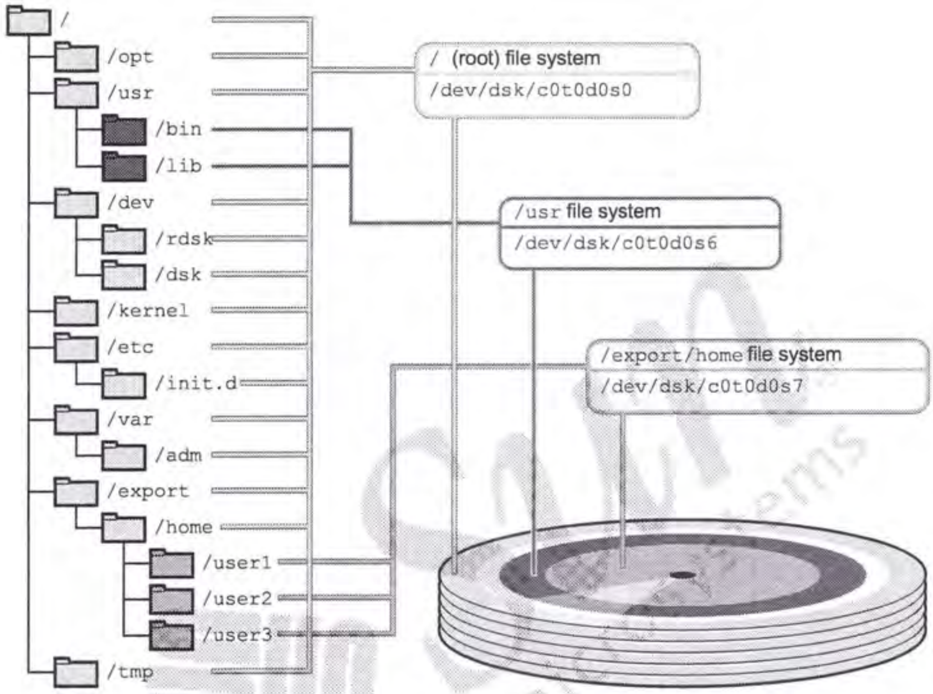

Figure 3-3 Solaris ufs File Systems Residing on Disk Slices

The Solaris OE ufs file system contains the following basic support structures.

Disk Label (VTOC)

The disk label (VTOC) contains the partition table for the disk. The VTOC resides in the first disk sector (512-byte blocks). Only the first disk slice contains a VTOC, although file systems created on any slice skip the first sector in case it might contain a VTOC.

Boot Block

The bootstrap program (`bootblk`) resides in the 15 disk sectors (Sectors 1–15) that follow the VTOC. Only the / (root) file system has an active boot block. However, space is allocated for a boot block at the beginning of each file system.

Primary Superblock

The superblock resides in the 16 disk sectors (Sectors 16–31) that follow the boot block. The superblock is a table of information that describes the file system, including:

- The number of data blocks
- The number of cylinder groups
- The size of a data block and fragment
- A description of the hardware, derived from the label
- The name of the mount point
- File system state flag: clean, stable, active, logging, or unknown

Backup Superblocks

When the file system is created, each cylinder group replicates the superblock beginning at Sector 32. This replication protects the critical data in the superblock against catastrophic loss.

Cylinder Groups

Each file system is divided into cylinder groups with a minimum default size of 16 cylinders per group. Cylinder groups improve disk access.

The file system constantly optimizes disk performance by attempting to place a file's data into a single cylinder group, which reduces the distance a head has to travel to access the file's data. The file system stores large files across several cylinder groups, if needed.

Cylinder Group Blocks

The cylinder group block is a table in each cylinder group that describes the cylinder group, including:

- The number of inodes
- The number of data blocks in the cylinder group
- The number of directories
- Free blocks, free inodes, and free fragments in the cylinder group
- The free block map
- The used inode map

Figure 3-4 shows a series of cylinder groups in a ufs file system.

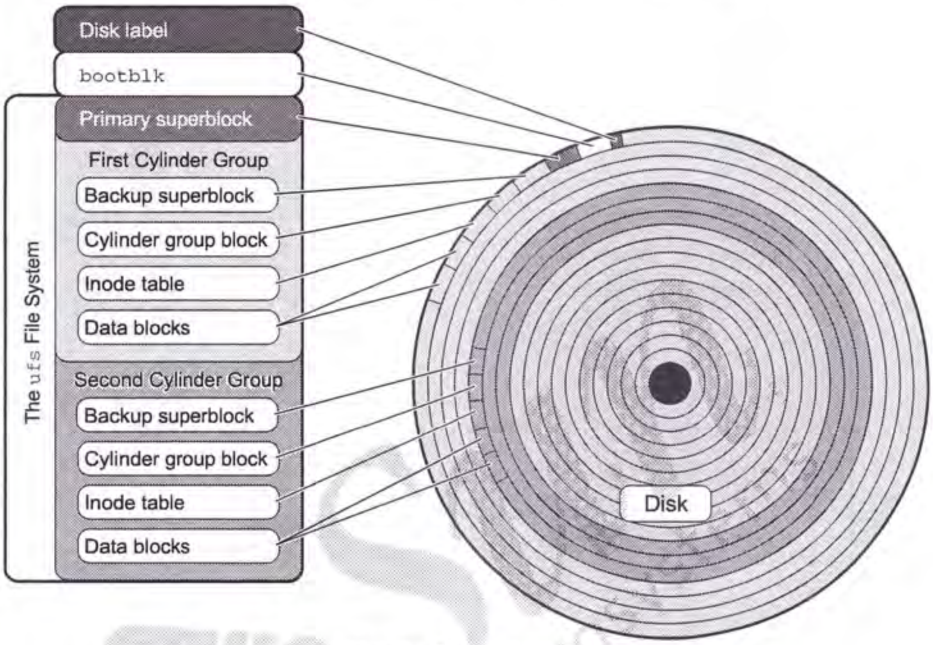

Figure 3-4 Solaris ufs File System Structure

The ufs Inode

An inode contains the following information about a file:

- The type of file and the access modes
- The user identification (UID) and group identification (GID) numbers of the file's owner and group
- The size of the file
- The link count
- The time the file was last accessed and modified and the inode changed
- The total number of data blocks used by or allocated to the file
- Two types of pointers: direct pointers and indirect pointers

Figure 3-5 shows some of the information contained in an inode.

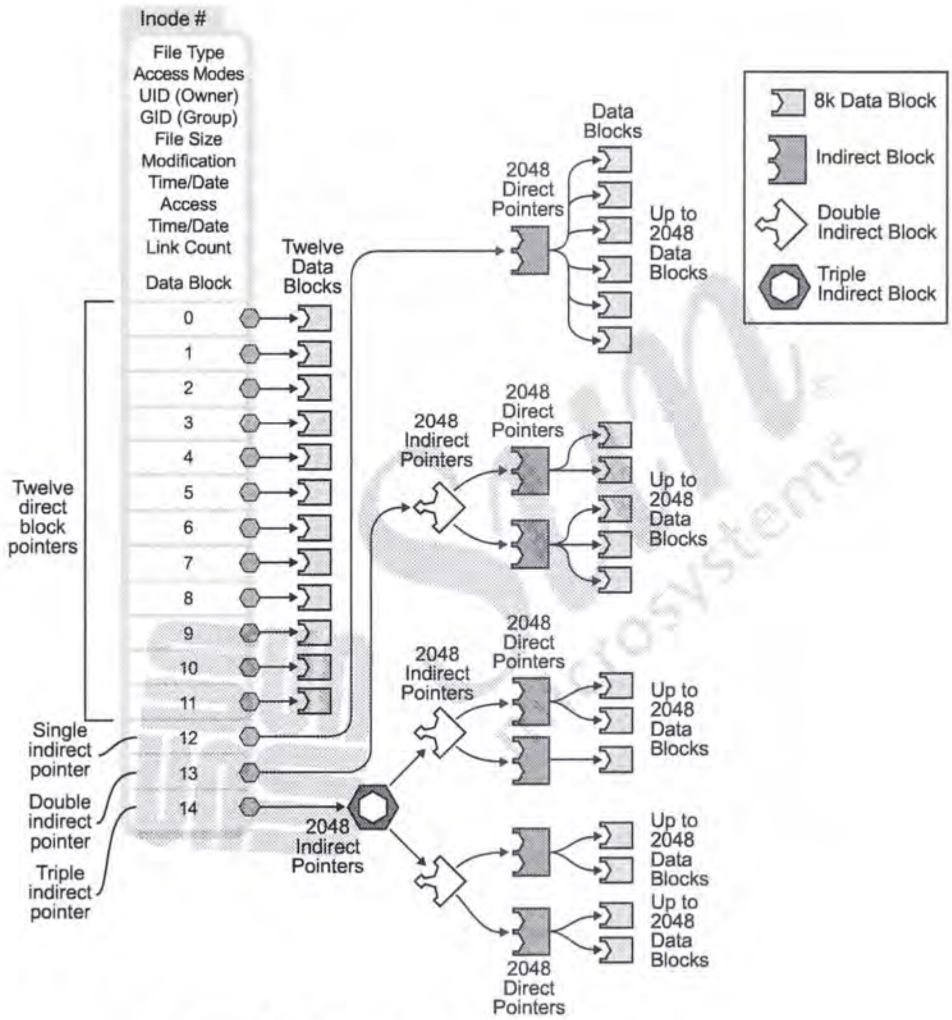

Figure 3-5 Structure of a ufs Inode

Note – To view some of the information contained in a file or directory inode, use the `ls -l` command. To view the inode number assigned to the file or directory, use the `ls -i` command.

Direct Pointers

Inside the inode there are 12 direct pointers, which contain addresses for the file's first 12 data blocks. The 12 direct pointers can each reference 8-Kbyte data blocks for a file that is up to 96 Kbytes.

Indirect Pointers

The three types of indirect pointers within an inode are:

- Single indirect pointer – Refers to a file system block that contains pointers to data blocks. This file system block contains 2048 additional addresses of 8-Kbyte data blocks, which can point to an additional 16 Mbytes of data.
- Double indirect pointer – Refers to a file system block that contains single indirect pointers. Each indirect pointer refers to a file system block that contains the data block pointers. Double indirect pointers point to an additional 32 Gbytes of data.
- Triple indirect pointer – Can reference up to an additional 64 Tbytes of data. However, the maximum size of a `ufs` file system is limited to 1 Tbyte due to the maximum address space of 32-bits for the device drivers.

Data Blocks

The remaining space allocated to the `ufs` file system holds data blocks. Data blocks are allocated, by default, in 8-Kbyte logical block sizes. The blocks are further divided into 1-Kbyte fragments. For a regular file, the data blocks contain the contents of the file. For a directory, the data blocks contain entries that associate the inode numbers and the file names of the files and directories contained in that directory.

Within a file system, those blocks that are currently not being used as files, directories, indirect address blocks, or storage blocks are marked as free in the cylinder group map. This map also keeps track of fragments to prevent disk performance from degrading.

Fragmentation

Fragmentation is the method used by the `ufs` file system to allocate disk space efficiently. Files less than 96 Kbytes in size are stored using fragmentation.

By default, data blocks can be divided into eight fragments of 1024 bytes each. Fragments store files and pieces of files smaller than 8192 bytes. For files larger than 96 Kbytes, fragments are never allocated and full blocks are exclusively used.

Figure 3-6 shows a fragment in a data block.

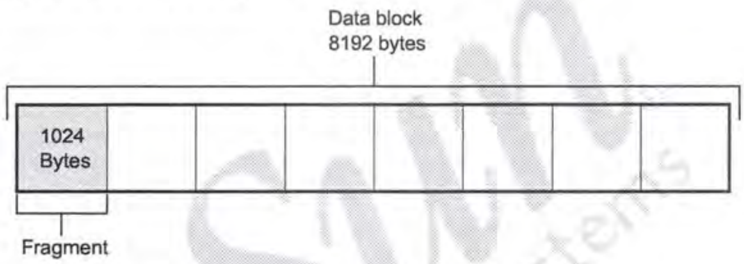

Figure 3-6 Divided Data Block

If a file contained in a fragment grows and requires more space, it is allocated one or more additional fragments in the same data block.

Figure 3-7 shows the contents of two different files stored in fragments in the same data block.

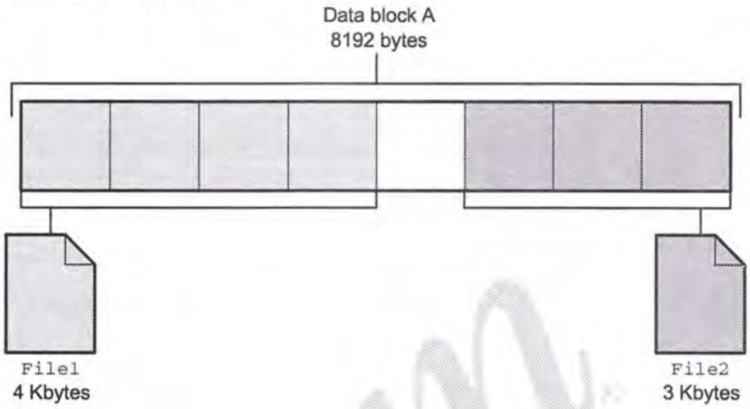

Figure 3-7 Two Files Stored in One Data Block

For example, if `File1` requires more space than is currently available in the shared data block, the entire contents of that expanding file are moved by the `ufs` file system into a free data block. This requirement by the `ufs` file system assures that all of a file's fragments are contained in a whole data block. The `ufs` file system does not allow fragments of the same file to be stored in two different data blocks.

Using the newfs Command

To use the disk to store directories or files, a file system must be created on every disk partition. As the root user, you can construct a ufs file system on a disk slice by using the newfs command.

The newfs command is an easy-to-use front-end to the mkfs command, which you use to create file systems. The newfs command is located in the /usr/sbin directory.

Caution – The newfs command is destructive. The command overwrites data that resides on the selected disk slice.

To create a ufs file system, by using the newfs command, perform the following steps:

1. As the root user, create a file system on a slice of a newly partitioned disk by entering the command:

```
# newfs /dev/rdsk/c0t0d0s7
```

2. The newfs command asks for confirmation before continuing. Verify that the correct disk slice on the correct disk is selected. To proceed, type y; to terminate the process, type n.

```
newfs: construct a new file system /dev/rdsk/c0t0d0s7: (y/n)? y
```

The newfs command displays information about the new file system being created.

```
/dev/rdsk/c0t0d0s7:     10510856 sectors in 2927 cylinders of 27 tracks,
133 sectors
        5132.3MB in 183 cyl groups (16 c/g, 28.05MB/g, 3392 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 57632, 115232, 172832, 230432, 288032, 345632, 403232, 460832,
518432,576032, 633632, 691232, 748832, 806432, 864032, 921632, 979232
(output omitted)
#
```

The first line printed by the newfs command describes the basic disk geometry. The second line describes the ufs file system created in this slice. The third and remaining lines list the beginning sector locations of the backup superblocks.

Note – This process also creates a lost+found directory for the ufs file system, which is a directory that is used by the file system check and repair (fsck) utility.

3. Repeat Steps 1 and 2 for every disk slice on any newly partitioned disk that needs to contain a file system.

The newfs command reserves between 1 and 10 percent of the file system space for maintenance of the file system. This free space, referred to as minfree, specifies the amount of space on the slice that is reserved or held back from regular users. You can use the newfs -m *%free* command to preset the percentage of free space when you create a new file system.

To show the value of minfree on a file system, use the fstyp command.

The following command shows the minfree value for the file system on the c0t0d0s0 device.

```
# fstyp -v /dev/rdsk/c0t0d0s0 | head -10
ufs
magic    11954   format  dynamic time     Fri Mar 15 07:53:40 2002
sblkno   16      cblkno  24      iblkno   32       dblkno  480
sbsize   3072    cgsize  2048    cgoffset 32       cgmask  0xfffffff0
ncg      17      size    121905  blocks   114000
bsize    8192    shift   13      mask     0xffffe000
fsize    1024    shift   10      mask     0xfffffc00
frag     8       shift   3       fsbtodb  1
minfree 10%      maxbpg  2048    optim    time
maxcontig 16     rotdelay 0ms    rps      90
```

To change the minimum percentage value of free space on an existing file system, you can use the tunefs -m *%free* command.

The following command changes the minimum percentage of free space on the /dev/rdsk/c0t0d0s0 device to 1 percent.

```
# tunefs -m 1 /dev/rdsk/c0t0d0s0
minimum percentage of free space changes from 10% to 1%
```

Checking the File System by Using the `fsck` Command

A file system can become damaged if it is corrupted from a power failure, a software error in the `kernel`, a hardware failure, or an improper shutdown of the system. The file system check program, `fsck`, checks the data consistency of a file system and attempts to correct or repair any inconsistencies or damage found.

Caution – Never run the `fsck` command on a mounted file system. This could leave the file system in an unusable state. It could also delete data. The / (root), `/usr`, and `/var` file systems should have the `fsck` command run on them while in single-user mode.

Every time you boot a system, the operating system determines which file systems the `fsck` command should check. The `fsck` command checks and repairs any problems encountered in file systems before they are mounted.

Note – The status of a file system's state flag determines whether the file system needs to be scanned by the `fsck` command. When the state flag is "clean," "stable," or "logging," file system scans are not run.

Data Inconsistencies Checked by the `fsck` Command

The `fsck` command makes several passes through a file system. During each pass, the `fsck` command checks for several types of file system inconsistencies.

Superblock Consistency

The file system superblock is checked for inconsistencies involving such parameters as file system size, free block count, and free inode count.

Cylinder Group Block Consistency

The `fsck` command checks any unallocated data blocks claimed by inodes, the unallocated data block count, and the unallocated inode count.

Inode Consistency

The `fsck` command checks for the allocation state of inodes, as well as the type, the link count, duplicate blocks (blocks already claimed by another inode), bad blocks, the inode size, and the block count for each inode. Any unreferenced inode with a nonzero link count is linked to the file system's `lost+found` directory.

Data Block Consistency

The `fsck` command cannot check ordinary data blocks, but it can check directory data blocks. In directory data blocks, the `fsck` command checks for inodes that point to unallocated blocks, unallocated blocks tagged as in use, allocated blocks tagged as free (incorrect inodes for `.` and `..`) and directories not connected to the file system. These directories are linked back to the file system in its `lost+found` directory.

The `lost+found` Directory

The `fsck` command puts files and directories that are allocated but unreferenced in the `lost+found` directory located in that file system. The inode number of each file is assigned as the file name. If the `lost+found` directory does not exist, the `fsck` command creates it. If not enough space exists in the `lost+found` directory, the `fsck` command increases the directory's size.

Noninteractive Mode

During a normal system boot, the `fsck` command operates in noninteractive mode, which is often referred to as preen, or silent mode. In this mode, the `fsck` command addresses only minor inconsistency problems that can be corrected. If a more serious inconsistency is found and a decision has to be made, the `fsck` program terminates and requests the `root` password to enter single-user mode. Execute the `fsck` command in interactive mode to continue.

Interactive Mode

In interactive mode, the fsck command lists each problem it encounters, followed by a suggested corrective action in the form of a question that requires a yes or no response.

The following example shows how the fsck command displays a message that asks if you want to correct the block count.

```
# fsck /dev/rdsk/c0t0d0s7
** /dev/rdsk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
INCORRECT BLOCK COUNT I=743 (5 should be 2)
CORRECT?
```

If you respond with yes, the fsck command applies the corrective action and moves on. If you respond with no, the fsck command repeats the message about the original problem and suggests corrective action. It does not fix the inconsistency until you respond yes.

The following examples demonstrate how you as the system's root user can run the fsck command to check the integrity of file systems.

- To check a single unmounted file system, perform the command:

```
# fsck /dev/rdsk/c0t0d0s7
```

  This is the only way to check a file system that has not been entered in the /etc/vfstab file.

- To check a file system using the mount point directory name as listed in the /etc/vfstab file, perform the command:

```
# fsck /export/home
```

In the following example, the fsck command checks and repairs the file system with the force (f) and preen (p) options.

```
# fsck -o f,p /dev/rdsk/c0t0d0s7
/dev/rdsk/c0t0d0s7: 77 files, 9621 used, 46089 free
/dev/rdsk/c0t0d0s7: (4 frags, 57 blocks, 0.0% fragmentation)
```

The f option of the fsck command forces a file system check, regardless of the state of the file system's superblock state flag.

The p option checks and fixes the file system noninteractively (preen). The program exits immediately if a problem requiring intervention is found.

Resolving File System Inconsistencies

If problems are located in a file system, you are alerted by the `fsck` utility. Some of the more common file system errors that require interactive intervention are:

- Allocated unreferenced file
- Inconsistent link count
- Free block count corruption
- Superblock corruption

Reconnecting an Allocated Unreferenced File

If the `fsck` command discovers an inode that is allocated but unreferenced or not linked in any directory, the command sends a message that asks you if you want to reconnect the inode.

```
** Phase 3 - Check Connectivity
UNREF FILE I=788 OWNER=root MODE=100644
SIZE=19994 MTIME=Jan 18 10:49 1999
RECONNECT? y
```

A yes response causes the `fsck` command to save the file to the `lost+found` directory. The `fsck` command references the inode number.

To determine the type of file moved to the `lost+found` directory by the `fsck` command, perform the following steps:

1. List the contents of the file system's `lost+found` directory.

```
# ls /export/home/lost+found
  #788
```

2. Determine the file type by using the `file` command.

```
# file /export/home/lost+found/#788
/export/home/lost+found/#788: ascii text
```

3. To view the contents of an ASCII text file, use the `more` or `cat` command. To view the contents of a binary file, use the `strings` command. If the file is associated with an application, such as a word processing document, use the application to view the contents of the file.

```
# cat /export/home/lost+found/#788
```

4. If the file is intact and you know where it belongs, you can copy the file back to its original location in the file system.

```
# cp /export/home/lost+found/#788 /export/home/user1/report
```

Adjusting a Link Counter

If the `fsck` program discovers that the value of a directory inode link counter and the actual number of directory links are inconsistent, the command displays a message that asks you if you want to adjust the counter.

```
** Phase 4 - Check Reference Counts
LINK COUNT DIR I=2 OWNER=root MODE=40755
SIZE=512 MTIME=Jan 18 15:59 1999 COUNT 4 SHOULD BE 3
ADJUST? y
```

In the example, a **y** (yes) response causes the `fsck` command to correct the directory inode link counter from 4 to 3.

During this phase, you might also be asked to clear or remove a link.

```
BAD/DUP type I=200 OWNER=root MODE=40755
SIZE=512 MTIME=Mar 14 08:03 2002
CLEAR? y
```

Salvaging the Free List

If the `fsck` utility discovers that the unallocated block count and the free block number listed in the superblock are inconsistent, the `fsck` command displays a message that asks if you want to salvage the free block count by rectifying it with the unallocated block count.

```
** Phase 5 - Check Cyl groups
CG 0: BAD MAGIC NUMBER
FREE BLK COUNT(S) WRONG IN SUPERBLK
SALVAGE? y
```

In the example, a **y** (yes) response causes the `fsck` command to update the information in the file system superblock.

Using Backup Superblocks

Superblock corruption can cause a file system to be unmountable. A file system is unusable when the message such as "`Can't mount` *`file_system_name`*" or "*`device_name`* `is not this fstype`" appears.

```
Can't mount /dev/dsk/c0t0d0s7
```

This message can appear during a system boot or when you are manually mounting the file system.

If the `fsck` command fails because of a corrupted superblock, you see an error message that tells you to execute the `fsck` command using a superblock backup to recover the file system. Execute the `fsck` command with the `-o` option and with the `b` flag followed by a backup superblock number. Every file system has an alternative backup superblock at block number 32, which can be used with the `fsck` command to repair the primary superblock.

The following command uses a backup superblock.

```
# fsck -o b=32 /dev/rdsk/c0t0d0s7
Alternate super block location: 32.
** /dev/rdsk/c0t0d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 9 used, 5174880 free (16 frags, 646858 blocks, 0.0%
fragmentation)
#
```

The `fsck` utility compares the information in the backup superblock with the actual file system and attempts to rebuild the primary superblock. However, if the first backup superblock is part of the file system that was damaged, it may be unusable. Select another backup superblock to continue the `fsck` command.

To list the locations of all the alternative backup superblocks in the file system, run the `newfs -N` command.

Caution – This method works if the underlying file system was built using the `newfs` default parameters. If the file system was not built with these defaults, execute the `newfs -N` command, using the same parameters originally used, to generate identical superblock locations.

Use the `-N` option to view the file system parameters that you could use to create a new file system without actually creating the file system. A portion of the output is a list of the locations of all the alternative backup superblocks that can be used with the `fsck -o b=#` command.

```
# newfs -N /dev/rdsk/c0t0d0s7
/dev/rdsk/c0t0d0s7:      10510856 sectors in 2927 cylinders of 27 tracks,
133 sectors
        5132.3MB in 183 cyl groups (16 c/g, 28.05MB/g, 3392 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 57632, 115232, 172832, 230432, 288032, 345632, 403232, 460832,
518432,576032, 633632, 691232, 748832, 806432, 864032, 921632, 979232
(output truncated)
#
```

You can use any other alternative superblock number in the list with the `fsck` command.

```
# fsck -o b=535952 /dev/rdsk/c0t0d0s7
Alternate super block location: 518432
** /dev/rdsk/c0t0d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
7 files, 14 used, 279825 free (17 frags, 347891 blocks, 0.0% fragmentation)
*****FILE SYSTEM WAS MODIFIED*****
#
```

Monitoring File System Use

An important activity of a system administrator is to monitor file system use on a regular basis. There are three useful commands available for this task:

- `df` – Displays the number of free disk blocks
- `du` – Summarizes disk use
- `quot` – Summarizes file system ownership

Using the `df` Command

Use the `df` command to display the amount of disk space used in file systems. This command lists the amount of used and available space and the amount of the file system's total capacity being used.

The format for the `df` command is:

```
df -option mount_point
```

Table 3-1 lists some of the more common options used with the `df` command.

Table 3-1 Partial Listing of Options for the `df` Command

Option	Description
`-a`	Reports on all file systems, including those with entries in the `/etc/mnttab` file for which the ignore option is set
`-b`	Prints the total number of Kbytes free
`-e`	Prints only the number of files free
`-k`	Displays disk allocation in Kbytes
`-h`	Acts like the `-k` option, except that sizes are in a more readable format, for example, `14K`, `234M`, `2.7G`, or `3.0T`
`-l`	Reports on local file systems only

To display the capacity of file systems, perform the command:

```
# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0     114000   69446   33154    68%    /
/dev/dsk/c0t0d0s6    1280269  990963  238096    81%    /usr
/proc                      0       0       0     0%    /proc
mnttab                     0       0       0     0%    /etc/mnttab
fd                         0       0       0     0%    /dev/fd
/dev/dsk/c0t0d0s1      54861   45062    4313    92%    /var
swap                  683232      48  683184     1%    /var/run
swap                  683184       0  683184     0%    /tmp
/dev/dsk/c0t0d0s5      24114     550   21153     3%    /opt
/dev/dsk/c0t0d0s7    2101887      92 2038739     1%    /export/home
```

The same file system displayed with the -h option would appear in human-readable format.

```
# df -h
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0t0d0s0      111M    68M    32M    68%    /
/dev/dsk/c0t0d0s6      1.2G   968M   233M    81%    /usr
/proc                    0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
fd                       0K     0K     0K     0%    /dev/fd
/dev/dsk/c0t0d0s1       54M    44M   4.2M    92%    /var
swap                   667M    48K   667M     1%    /var/run
swap                   667M     0K   667M     0%    /tmp
/dev/dsk/c0t0d0s5       24M   550K    21M     3%    /opt
/dev/dsk/c0t0d0s7      2.0G    92K   1.9G     1%    /export/home
```

Table 3-2 defines the fields displayed by the `df -k` command.

Table 3-2 Fields for the `df -k` Command

Field	Definition
`Filesystem`	The mounted file system
`kbytes`	The size of the file system in Kbytes (1024 bytes)
`used`	The number of Kbytes used
`avail`	The number of Kbytes available
`capacity`	The percentage of file system capacity used
`Mounted on`	The mount point

The amount of space that is reported as `used` and `avail` is typically less than the amount of total space in the file system. A fraction of space, from 1 to 10 percent, is reserved in each file system as the `minfree` value.

When all of the reported space on the file system is in use, the file system capacity is displayed as 100 percent. Regular users receive the message "`File System Full`" and cannot continue working. The reserved space is available to the `root` user, who can then delete or back up files in the file system.

Using the du Command

Use the `du` command to display the number of disk blocks used by directories and files. Each disk block consists of 512 bytes.

The format for the `du` command is:

```
du -options directory
```

Table 3-3 describes the options for the `du` command.

Table 3-3 Options for the `du` Command

Option	Description
`-k`	Displays disk use in Kbytes.
`-s`	Displays only the summary in 512-byte blocks. Using the `s` and `k` options together shows the summary in Kbytes.
`-a`	Displays the number of blocks used by all files in addition to directories within the specified directory hierarchy.

To display disk usage in kilobytes, perform the command:

```
# cd /opt
# du -k
8       ./lost+found
3       ./SUNWits/Graphics-sw/xil/lib
4       ./SUNWits/Graphics-sw/xil
7       ./SUNWrtvc/man/man1
(some output removed for brevity)
19      ./SUNWrtvc/man/man3
27      ./SUNWrtvc/man
535     ./SUNWrtvc
550     .
```

To display disk usage including files, perform the command:

```
# du -ak /opt
8       /opt/lost+found
1       /opt/SUNWits/Graphics-sw/xil/lib/libxil.so
1       /opt/SUNWits/Graphics-sw/xil/lib/libxil.so.1
3       /opt/SUNWits/Graphics-sw/xil/lib
(some output removed for brevity)
27      /opt/SUNWrtvc/man
535     /opt/SUNWrtvc
550     /opt
```

To display only a summary of disk usage, perform the command:

```
# du -sk /opt
550     /opt
```

Using the `quot` Command

Use the `quot` command to display how much disk space, in kilobytes, is being used by users.

Note – The `quot` command can be run only by the `root` user.

The format for the `quot` command is:

```
quot -options filesystem
```

Table 3-4 describes the options for the `quot` command.

Table 3-4 Options for the `quot` Command

Option	Description
`-a`	Reports on all mounted file systems
`-f`	Includes the number of files

To display disk space being used by users on all mounted file systems, perform the command:

```
# quot -af
/dev/rdsk/c0t0d0s0 (/):
     14326      1284     root
      4792        37     bin
        31        27     lp
         1         1     sys
/dev/rdsk/c0t0d0s6 (/usr):
    197394      6962     root
    161203     11884     bin
      2140       232     lp
         1         1     adm
```

The columns represent kilobytes used, number of files, and owner, respectively.

To display a count of the number of files and space owned by each user for a specific file system, enter the following:

```
# quot -f /dev/dsk/c0t0d0s7
/dev/dsk/c0t0d0s7:
    134      62     root
    103      84     user1
    140      32     user9
```

Using the Solaris Management Console Usage Tool

The Solaris Management Console Usage Tool provides a graphical display of the available space for all mounted file systems.

To use the Solaris Management Console storage Usage Tool, launch the Solaris Management Console by typing **smc&** at a command line, or select it from the Application Manager Window. To locate the Usage Tool, select This Computer, then select Storage, then select Mounts and Shares on the Solaris Management Console.

Figure 3-8 shows the Management Tools: Solaris Management Console window with the disk usage information.

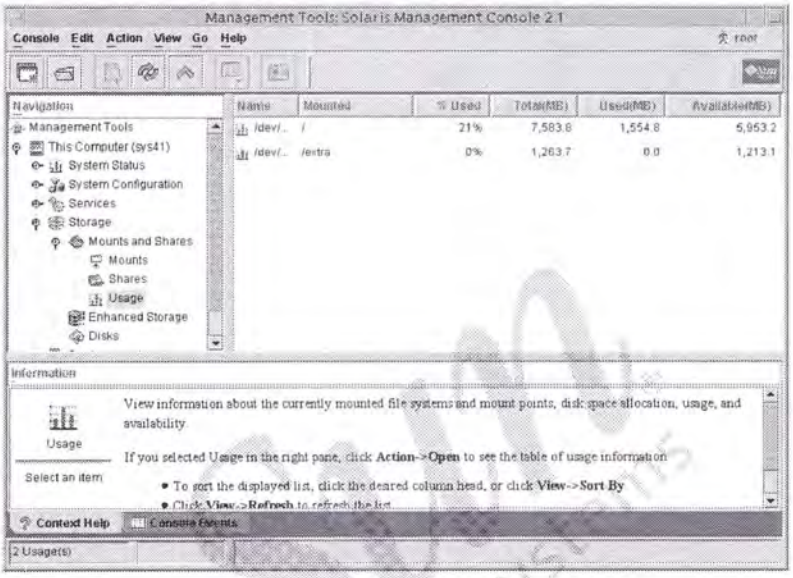

Figure 3-8 Management Tools: Solaris Management Console Window

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Creating and Maintaining `ufs` File Systems (Level 1)

In this exercise, you complete the following tasks:

- Create `ufs` file systems
- Calculate and adjust `minfree` values
- Destroy the superblock on an unused file system and repair it using an alternative

Preparation

This exercise requires an unused disk divided into four slices. Slices 0, 1, and 3 are equal in size, and Slice 4 takes up the remaining space on the disk. If it is necessary to partition the disk, this exercise requires an understanding of how to use the `format` utility. Refer to the lecture notes as necessary to perform the steps.

Tasks

Perform the following tasks:

- Find a disk that is not in use and that is partitioned as specified in the preceding preparation description. If necessary, partition a disk accordingly. Create a new file system on Slice 0. Create a file system on Slice 1 with an inode ratio of 1 per 16,384 bytes of data space. Compare how quickly the `newfs` command makes the file systems. For both file systems, record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group. How do the file systems differ?

  (Steps 1–6 in the Level 2 lab)

- Display the number of Kbytes used, the number available, and the number allocated to both file systems. Record these values. Which file system has more available space and why? For each file system, calculate how much larger the Kbytes value is than the sum of the used and available values, and express the result as a percentage. Use the `fstyp` command to verify the result.

  (Steps 7–8 in the Level 2 lab)

- Adjust the `minfree` value up or down by 3 percent. Record the message that your command displays. Verify the change made by using the `tunefs` command.

  (Steps 9–10 in the Level 2 lab)

- Create new file systems on Slices 3 and 4 of your spare disk.

  (Step 11 in the Level 2 lab)

- Check the file system on Slice 3 with the `fsck` command, and record if it reports any errors. Use the `dd` command from Step 13 in the Level 2 lab to destroy the primary superblock of the new file system. Run the `fsck` command, and see if you get an error. Use the `fsck` command and the backup superblock found at Sector 32 to repair the file system and main superblock. Verify the repair by running the `fsck` command again.

  (Steps 12–16 in the Level 2 lab)

Exercise: Creating and Maintaining `ufs` File Systems (Level 2)

In this exercise, you complete the following tasks:

- Create `ufs` file systems
- Calculate and adjust `minfree` values
- Destroy the superblock on an unused file system and repair it using an alternative

Preparation

This exercise requires an unused disk, divided into four slices. Slices 0, 1, and 3 are equal in size, and Slice 4 takes up the remaining space on the disk. If it is necessary to partition this disk, this exercise requires an understanding of how to use the `format` utility. Refer to the lecture notes as necessary to perform the steps.

Task Summary

In this exercise, you accomplish the following:

- Find a disk that is not in use and that is partitioned as specified in the preceding preparation description. If necessary, partition a disk accordingly. Create a new file system on Slice 0. Create a file system on Slice 1 with an inode ratio of 1 per 16,384 bytes of data space. Compare how quickly the `newfs` command makes the file systems. For both file systems, record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group. How do the file systems differ?
- Display the number of Kbytes used, the number available, and the number allocated to both file systems. Record these values. Which file system has more available space and why? For each file system, calculate how much larger the Kbytes value is than the sum of the used and available values, and express the result as a percentage. Use the `fstyp` command to verify the result.
- Adjust the `minfree` value up or down by 3 percent. Record the message that your command displays. Verify the change made by using the `tunefs` command.

- Create new file systems on Slices 3 and 4 of your spare disk.
- Check the file system on Slice 3 with the `fsck` command, and record if it reports any errors. Use the `dd` command from Step 13 in the Level 2 lab to destroy the primary superblock of the new file system. Run the `fsck` command, and see if you get an error. Use the `fsck` command and the backup superblock found at Sector 32 to repair the file system and main superblock. Verify the repair by running the `fsck` command again.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Change the directory to `/dev/rdsk`.
2. To find a spare disk, use the `ls` command to display a list of possible disks and the `prtvtoc` command to display the VTOC for each disk you find. Examine the partition list as well as the `Mount Directory` field that the `prtvtoc` command displays. Disks that are not in use have no mount directory listed. Record the name of the unused disk.

   Unused disk:

Note – This procedure works for the classroom environment. A disk that does not show mounted slices in the Mount Directory field of the `prtvtoc` output is not necessarily inactive.

---

3. If a spare disk exists but it is not divided into four slices, use the `format` utility to partition the disk. Make three slices exactly the same size (approximately 25 percent of the total disk space each), and use the fourth partition for the remainder of the available space. Exit from the `format` utility when you are finished. You can also use the Solaris Management Console to partition the drive.
4. Use the `newfs` command without options to create a new file system on Slice 0 on the spare disk. Observe how quickly the `newfs` command creates cylinder groups on this slice. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

   Cylinder groups:

   Cylinders per group:

   Inodes per group:

5. Use the newfs command to create a new file system on Slice 1 on the spare disk. Use the -i option to create one inode per 16,384 bytes of data space. Observe how quickly the newfs command creates cylinder groups on this slice. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

   Cylinder groups:

   Cylinders per group:

   Inodes per group:

6. According to the statistics you have gathered, how do the file systems on Slices 0 and 1 differ?

7. Use the df command to display statistics for the file systems on Slices 0 and 1 that you used in the previous steps. Record the values listed in the kbytes, used, and avail columns.

   Which file system has the larger amount of available data space and why?

8. For each file system, add the used and avail values, and compare the sum to the kbytes value. Expressed as a percentage, how much larger is the kbytes value than the sum of used and avail? This percentage should approximately match the minfree value.

   Use the fstyp command to verify your result.

9. Use the tunefs command to change the minfree value for the file system on Slice 0 of the spare disk. If the current minfree value is greater than 5 percent, reduce it by 3 percent. If it is less than or equal to 5 percent, add 3 percent.

   What message does the tunefs command display?

10. Use the df -k command to verify that the minfree value has changed. Record the values listed in the kbytes, used, and avail columns.

    Which value has changed from the information you gathered in Step 7?

11. Create new file systems on Slices 3 and 4 of your spare disk.

12. Run the fsck command interactively to check the new file system previously created on Slice 3 of the spare disk.

    Did the fsck command report errors?

13. Use the dd command to destroy the main superblock of the file system on Slice 3.

```
# dd if=/dev/zero of=/dev/rdsk/c1t0d0s3 count=32 bs=512
```

14. Run the fsck command interactively to check the new file system.

    Did the fsck command report errors? If so, what corrective action does the fsck command suggest?

15. Run the fsck command, and specify an alternative superblock. Block 32 is always one of the alternatives available.

16. Run the fsck command again to verify that the file system was repaired.

Exercise: Creating and Maintaining `ufs` File Systems (Level 3)

In this exercise, you complete the following tasks:

- Create `ufs` file systems
- Calculate and adjust `minfree` values
- Destroy the superblock on an unused file system and repair it using an alternative

Preparation

This exercise requires an unused disk, divided into four slices. Slices 0, 1, and 3 are equal in size, and Slice 4 takes up the remaining space on the disk. If it is necessary to partition this disk, this exercise requires an understanding of how to use the `format` utility. Refer to the lecture notes as necessary to perform the steps.

Task Summary

In this exercise, you accomplish the following:

- Find a disk that is not in use and that is partitioned as specified in the preceding preparation description. If necessary, partition a disk accordingly. Create a new file system on Slice 0. Create a file system on Slice 1 with an inode ratio of 1 per 16,384 bytes of data space. Compare how quickly the `newfs` command makes the file systems. For both file systems, record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group. How do the file systems differ?
- Display the number of Kbytes used, the number available, and the number allocated to both file systems. Record these values. Which file system has more available space and why? For each file system, calculate how much larger the Kbytes value is than the sum of the used and available values, and express the result as a percentage. Use the `fstyp` command to verify the result.
- Adjust the `minfree` value up or down by 3 percent. Record the message that your command displays. Verify the change made by using the `tunefs` command.

- Create new file systems on Slices 3 and 4 of your spare disk.
- Check the file system on Slice 3 with the fsck command, and record if it reports any errors. Use the dd command from Step 13 in the Level 2 lab to destroy the primary superblock of the new file system. Run the fsck command, and see if you get an error. Use the fsck command and the backup superblock found at Sector 32 to repair the file system and main superblock. Verify the repair by running the fsck command again.

Tasks and Solutions

Complete the following steps:

1. Log in as the root user, and open a terminal window. Change the directory to /dev/rdsk.

```
# cd /dev/rdsk
```

2. To find a spare disk, use the ls command to display a list of possible disks and the prtvtoc command to display the VTOC for each disk you find. Examine the partition list as well as the Mount Directory field that the prtvtoc command displays. Disks that are not in use have no mount directory listed. Record the name of the unused disk.

```
# ls *s2
# prtvtoc /dev/rdsk/c1t0d0s2
```

Unused disk:

Note – This procedure works for the classroom environment. A disk that does not show mounted slices in the Mount Directory field of the prtvtoc output is not necessarily inactive.

3. If a spare disk exists, but it is not divided into four slices, use the format utility to partition the disk. Make three slices exactly the same size (approximately 25 percent of the total disk space each), and use the fourth partition for the remainder of the available space. You can also use the Solaris Management Console to partition the drive. Exit from the format utility when you are finished.

Example of the partition table:

```
Part      Tag    Flag     Cylinders        Size            Blocks
  0 alternates    wm       0 - 1168        2.00GB    (1169/0/0)   4197879
  1 alternates    wm    1169 - 2337        2.00GB    (1169/0/0)   4197879
  2     backup    wm       0 - 4923        8.43GB    (4924/0/0)  17682084
  3 alternates    wm    2338 - 3506        2.00GB    (1169/0/0)   4197879
  4 alternates    wm    3507 - 4922        2.42GB    (1416/0/0)   5084856
```

4. Use the newfs command without options to create a new file system on Slice 0 on the spare disk. Observe how quickly the newfs command creates cylinder groups on this slice. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

```
# newfs /dev/rdsk/c1t0d0s0
```

Cylinder groups:

Cylinders per group:

Inodes per group:

5. Use the newfs command to create a new file system on Slice 1 on the spare disk. Use the -i option to create one inode per 16,384 bytes of data space. Observe how quickly the newfs command creates cylinder groups on this slice. Record the number of cylinder groups, the number of cylinders per group, and the number of inodes per group.

```
# newfs -i 16384 /dev/rdsk/c1t0d0s1
```

Cylinder groups:

Cylinders per group:

Inodes per group:

6. According to the statistics you have gathered, how do the file systems on Slices 0 and 1 differ?

The number of inodes per group is less on File System 1 than on File System 0.

7. Use the df command to display statistics for the file systems on Slices 0 and 1 that you used in the previous steps, for example:

```
# df -k /dev/dsk/c1t0d0s0
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c1t0d0s0    8705501       9 8618436     0%
# df -k /dev/dsk/c1t0d0s1
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c1t0d0s1    8769565       9 8681796     0%
```

Record the values listed in the kbytes, used, and avail columns.

Which file system has the larger amount of available data space and why?

File System 1 has the larger amount of available data space because it holds fewer inode records.

8. For each file system, add the `used` and `avail` values, and compare the sum to the `kbytes` value. Expressed as a percentage, how much larger is the `kbytes` value than the sum of `used` and `avail`? This percentage should approximately match the `minfree` value.

   Use the `fstyp -v /dev/rdsk/c#t#d#s# | head -12` command to verify your result.

   To calculate the percentage difference between the sum of `used` *and* `avail` *and the* `kbytes` *value, perform the following:*

   a. *Add the values listed as* `used` *and* `avail`*, for example:*

      9 + 1926799 = 1926808

   b. *Divide the sum of* `used` *and* `avail` *by the* `kbytes` *value, for example:*

      1926808 / 1986439 = 0.969981

   c. *Multiply the result of Step b by 100, for example:*

      0.969981 * 100 = 96.9981

   d. *Subtract the result of Step c from 100, for example:*

      100 - 96.9981 = 3.0019

   e. *Round the result of Step d to the nearest whole number, for example:*

      3.0019 = 3 percent

9. Use the `tunefs -m # /dev/rdsk/c#t#d#s#` command to change the `minfree` value for the file system on Slice 0 of the spare disk. If the current `minfree` value is greater than 5 percent, reduce it by 3 percent. If it is less than or equal to 5 percent, add 3 percent, for example:

```
# tunefs -m 4 /dev/rdsk/c1t0d0s0
minimum percentage of free space changes from 1% to 4%
```

   What message does the `tunefs` command display?

`The minimum percentage of free space changes from x percent to x percent.`

10. Use the `df -k` command to verify that the `minfree` value has changed. Record the values listed in the `kbytes`, `used`, and `avail` columns, for example:

```
# df -k /dev/dsk/c1t0d0s0
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c1t0d0s0    8705501       9 8357271     0%
```

Which value has changed from the information you gathered in Step 7?

The `avail` *column changes but not the* `kbytes` *or* `used` *columns.*

11. Create new file systems on Slices 3 and 4 of your spare disk, for example:

```
# newfs /dev/rdsk/c1t0d0s3
# newfs /dev/rdsk/c1t0d0s4
```

12. Run the `fsck` command interactively to check the new file system previously created on Slice 3 of the spare disk.

Did the `fsck` command report errors?

No.

13. Use the `dd` command to destroy the main superblock of the file system on Slice 3.

```
# dd if=/dev/zero of=/dev/rdsk/c1t0d0s3 count=32 bs=512
```

14. Run the `fsck` command interactively to check the new file system.

Did the `fsck` command report errors? If so, what corrective action does the `fsck` command suggest?

The `fsck` *command indicates that the magic number in the superblock is wrong and suggests repairing it by using an alternative superblock, for example:*

```
** /dev/rdsk/c1t0d0s3
BAD SUPER BLOCK: MAGIC NUMBER WRONG
USE AN ALTERNATE SUPER-BLOCK TO SUPPLY NEEDED INFORMATION;
e.g. fsck [-F ufs] -o b=# [special ...]
where # is the alternate super block. SEE fsck_ufs(1M).
```

15. Run the `fsck` command, and specify an alternative superblock. Block 32 is always one of the alternatives available.

```
# fsck -o b=32 /dev/rdsk/c1t0d0s3
```

16. Run the `fsck` command again to verify that the file system was repaired.

```
# fsck /dev/rdsk/c1t0d0s3
```

This time the `fsck` *command output does not report that the file system was modified.*

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Performing Mounts and Unmounts

Objectives

Upon completion of this module, you should be able to:

- Identify mounting fundamentals
- Perform mounts
- Perform unmounts
- Access a mounted diskette or CD-ROM
- Restrict access to a mounted diskette or CD-ROM
- Access a diskette or CD-ROM without Volume Management

The following course map shows how this module fits into the current instructional goal.

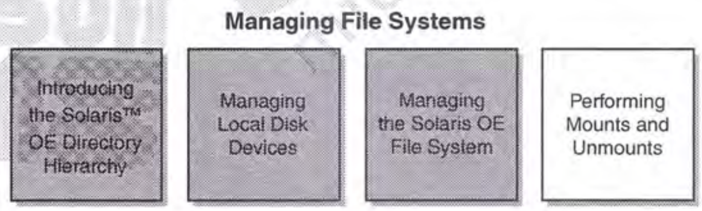

Figure 4-1 Course Map

Working With Mounting Fundamentals

In the Solaris OE, you use the mounting process to attach individual file systems to their mount points on the directory hierarchy. This action makes a file system accessible to the system and to the users.

You use the unmounting process to detach a file system from its mount point in the directory hierarchy. This action makes a file system unavailable to the system or users.

After you have created a file system by using the `newfs` command, you must attach it to the Solaris OE directory hierarchy at a mount point. A mount point is a directory that is the point of connection for a file system. File systems are commonly referred to by the names of their mount points, for example, the / (root) file system or the `/usr` file system.

Figure 4-2 shows how the directory hierarchy spans from one file system to the next.

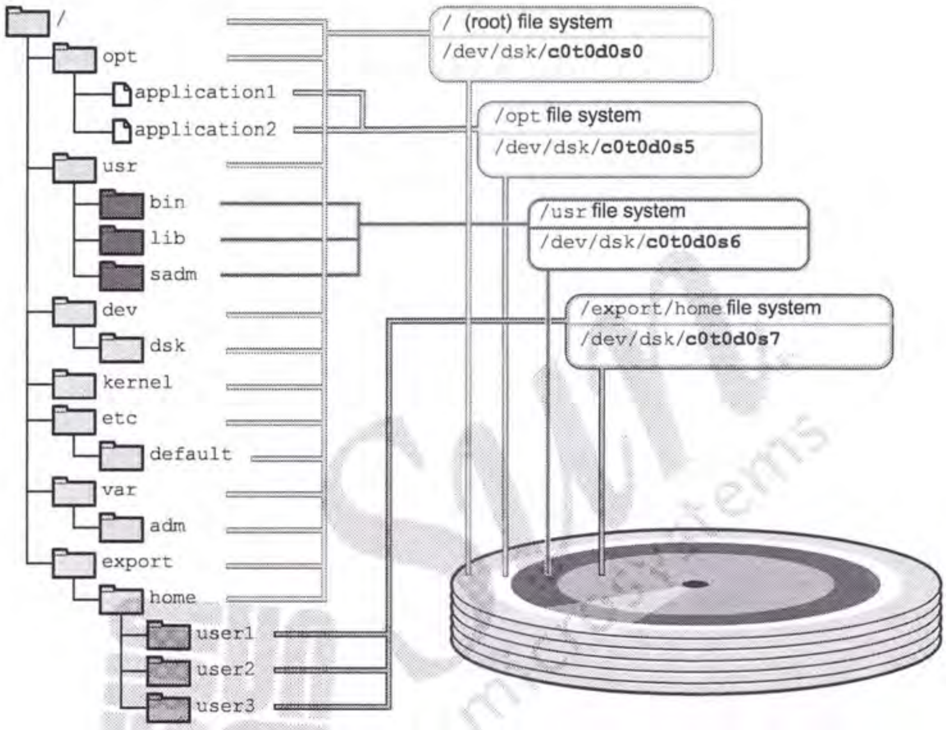

Figure 4-2 File Systems and Mount Points

File systems do not contain their own mount point directories.

Determining Which File Systems Are Currently Mounted

You can determine which file systems are currently mounted by using the `mount` command or the `df` command.

The `df` command displays the amount of disk space occupied by mounted or unmounted file systems and, depending on the options used, displays both locally mounted and virtual file system information.

The `mount` command, which is located in the `/usr/sbin` directory, maintains a table of currently mounted file systems in the `/etc/mnttab` file. When the `mount` command is used without arguments, it lists all of the mounted file systems in the `/etc/mnttab` directory. When used with only a partial argument list, the command searches the `/etc/vfstab` file for an entry that supplies the missing arguments.

Note – While system administrators typically use the `/usr/sbin/mount` command, the system boot scripts use the `/sbin/mount` command.

Mounting a File System Automatically

The Solaris OE provides several methods for automating file system mounts.

The Solaris OE creates a default `/etc/vfstab` file during software installation, based on your selections. However, you can edit the `/etc/vfstab` file whenever file system entries need to be added or modified.

Note – The automounter can mount network file systems on demand.

Introducing the Virtual File System Table: `/etc/vfstab`

The `/etc/vfstab` file lists all the file systems to be automatically mounted at system boot time, with the exception of the `/etc/mnttab` and `/var/run` file systems.

The file format includes seven fields per line entry. By default, a tab separates each field, but any whitespace can be used for separators. The dash (-) character is used as a placeholder for fields when text arguments are not appropriate. Commented lines begin with the hash (#) symbol.

To add a line entry, you need the following information:

`device to mount`	The device to be mounted. For example, a local `ufs` file system `/dev/dsk/c#t#d#s#`, or a pseudo file system `/proc`.
`device to fsck`	The raw or *character* device checked by the file system check program (`fsck`) if applicable. A pseudo file system has a dash (-) in this field.
`mount point`	The name of the directory that serves as the attach mount point in the Solaris OE directory hierarchy.
`FS type`	The type of file system to be mounted.
`fsck pass`	Indicates whether the file system is to be checked by the `fsck` utility at boot time. A 0 (zero) or a nonnumeric in this field indicates `no`. A 1 in this field indicates the `fsck` utility gets started for that entry and runs to completion. A number greater than 1 indicates that the device is added to the list of devices to have the `fsck` utility run. The `fsck` utility can run on up to eight devices in parallel. This field is ignored by the `mountall` command.
`mount at boot`	Enter `yes` to enable the `mountall` command to mount the file systems at boot time. Enter `no` to prevent a file system mount at boot time.

Note – For `/` (root), `/usr`, and `/var` (if it is a separate file system) file systems, the `mount at boot` field value is specified as `no`. The `kernel` mounts these file systems as part of the boot sequence before the `mountall` command is run. The `mount` command explicitly mounts the file systems `/` (root) and `/usr` as specified in the `/etc/rcS.d/S30rootusr.sh` script and the `/var` file system as specified in the `/etc/rcS.d/S70buildmnttab` script.

`mount options`	A comma-separated list of options passed to the `mount` command. A dash (-) indicates the use of default mount options.

Note – Because the default is to use tabs to separate the fields in the /etc/vfstab file, the fields often do not line up under their respective headings. This can lead to some confusion when you are viewing this file in a terminal window.

An example of a /etc/vfstab file follows:

```
# more /etc/vfstab
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
fd              -               /dev/fd         fd      -       no      -
/proc           -               /proc           proc    -       no      -
/dev/dsk/c0t0d0s1  -            -               swap    -       no      -
/dev/dsk/c0t0d0s0  /dev/rdsk/c0t0d0s0  /  ufs   1       no      -
/dev/dsk/c0t0d0s7  /dev/rdsk/c0t0d0s7  /export/home ufs  2  yes  -
swap    -       /tmp    tmpfs   -       yes     -
#
```

Introducing the /etc/mnttab File

The /etc/mnttab file is really an mntfs file system that provides read-only information directly from the kernel about mounted file systems on the local host.

Each time a file system is mounted, the mount command adds an entry to this file. Whenever a file system is unmounted, its entry is removed from the /etc/mnttab file.

Mount Point	The mount point or directory name where the file system is to be attached within the / (root) file system (for example, /usr, /opt).
Device Name	The name of the device that is mounted at the mount point. This block device is where the file system is physically located.
Mount Options	The list of mount options in effect for the file system.
dev=*number*	The major and minor device number of the mounted file system.
Date and Time Mounted	The date and time that the file system was mounted to the directory hierarchy.

The /var/run file system is a tmpfs mounted file system in the Solaris OE. It is the repository for temporary operating system files that are not needed across system reboots in this Solaris OE release. It is mounted as a pseudo file system rather than a disk-based file system.

The /var/run directory requires no administration. For security reasons, it is owned by the root user.

The /tmp directory continues to be a tmpfs mounted file system in the Solaris OE. It is the repository for temporary user and application files that are not needed across system reboots. It is a pseudo file system rather than a disk-based file system.

The following examples show two ways to display currently mounted file systems.

```
# more /etc/mnttab
/dev/dsk/c0t0d0s0       /       ufs
rw,intr,largefiles,xattr,onerror=panic,suid,dev=2200000 10164665
25
/proc   /proc   proc    dev=4300000     1016466524
mnttab  /etc/mnttab     mntfs   dev=43c0000     1016466524
fd      /dev/fd fd      rw,suid,dev=4400000     1016466527
swap    /var/run        tmpfs   xattr,dev=1     1016466529
swap    /tmp    tmpfs   xattr,dev=2     1016466532
/dev/dsk/c0t0d0s7       /export/home    ufs
rw,intr,largefiles,xattr,onerror=panic,suid,dev=2200007
1016466532
-hosts  /net    autofs  indirect,nosuid,ignore,nobrowse,dev=4580001
1016466537
auto_home       /home   autofs  indirect,ignore,nobrowse,dev=4580002
1016466537
-xfn    /xfn    autofs  indirect,ignore,dev=4580003     1016466537
sys41:vold(pid248)      /vol    nfs     ignore,dev=4540001      1016466554
```

```
# mount
/ on /dev/dsk/c0t0d0s0
read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200000 on Mon Mar 18
08:48:45 2002
/proc on /proc read/write/setuid/dev=4300000 on Mon Mar 18 08:48:44 2002
/etc/mnttab on mnttab read/write/setuid/dev=43c0000 on Mon Mar 18 08:48:44 2002
/dev/fd on fd read/write/setuid/dev=4400000 on Mon Mar 18 08:48:47 2002
/var/run on swap read/write/setuid/xattr/dev=1 on Mon Mar 18 08:48:49 2002
/tmp on swap read/write/setuid/xattr/dev=2 on Mon Mar 18 08:48:52 2002
/export/home on /dev/dsk/c0t0d0s7
read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200007 on Mon Mar 18
08:48:52 2002
```

Performing Mounts

You can mount file systems manually by logging in as the `root` user and running the `mount` command, or the system can automatically mount file systems at boot time after consulting the `/etc/vfstab` file.

Mounting a Local File System Manually

The `mount` command not only lists which file systems are currently mounted, it also provides you, as the `root` user, with a method for mounting file systems.

Default Behavior of the `mount` Command

To mount a local file system manually, you need to know the name of the device where the file system resides and its mount point directory name. Perform the command:

```
# mount /dev/dsk/c0t0d0s7 /export/home
```

In this example, the default action mounts the file system with the following options: `read/write`, `setuid`, `intr`, `nologging`, and `largefiles`, `xattr`, and `onerror`.

The following list explains the default options for the `mount` command.

`read/write`	Indicates whether reads and writes are allowed on the file system.
`setuid`	Permits the execution of `setuid` programs in the file system.
`intr/nointr`	Allows and forbids keyboard interrupts to kill a process that is waiting for an operation on a locked file system.
`nologging`	Indicates that logging is not enabled for the `ufs` file system.
`largefiles`	Allows for the creation of files larger than 2 Gbytes. A file system mounted with this option can contain files larger than 2 Gbytes.
`xattr`	Supports extended attributes not found in standard UNIX attributes.

Note – Due to file system overhead, the largest file size that can be created is approximately 866 Gbytes.

`onerror=`*`action`*	Specifies the action that the `ufs` file system should take to recover from an internal inconsistency on a file system. An action can be specified as: `panic` — Causes a forced system shutdown. This is the default. `lock` — Applies a file system lock to the file system. `umount` — Forcibly unmounts the file system.

The `/etc/vfstab` file provides you with another important feature. Because the `/etc/vfstab` file contains the mapping between the mount point and the actual device name, the `root` user can manually mount a file system specifying only the mount point on the command line.

```
# mount /export/home
```

Using the `mount` Command Options

When you are using `mount` options on the command line, remember that the options are preceded by the `-o` flag. When you are using multiple options, enter them as a comma-separated list following the `-o` flag.

```
mount -o option,option,... device_name mount_point
```

Note – Mount options are described in detail in the man page for the `mount_ufs` command.

Some options used to mount local file systems include: ro, nosetuid, noatime, nolargefiles, and logging.

- ro – Mounts the file system as read-only.

  The following is an example using this option on the command line:

```
# mount -o ro /dev/dsk/c0t0d0s6 /usr
```

- nosuid – Prohibits the execution of setuid programs in the file system. This does not restrict the creation of setuid programs.

  The following example shows the use of multiple options on the command line:

```
# mount -o ro,nosuid /dev/dsk/c0t0d0s7 /export/home
```

- noatime – Suppresses the time-last-accessed modification on inodes, which reduces disk activity on a file system where access times are not important. Specifying this option generally improves file access times and boosts overall performance, for example:

```
# mount -o noatime /dev/dsk/c0t0d0s7 /export/home
```

- nolargefiles – Prevents a file system that contains one or more "large files" from being mounted, for example:

```
# mount -o nolargefiles /dev/dsk/c0t0d0s7 /export/home
```

  Use of the nolargefiles option fails if the file system to be mounted contains a large file or did contain a large file at one time.

  If the file system currently contains a large file and the root user needs to mount it with this option, then the large file must be located and moved or removed from the file system. Then you must execute the fsck command manually to update the superblock information.

  The mount also fails if the file system at one time contained a large file, even though it was moved or removed. You must execute the fsck command to clear the old information and allow the file system to be mounted.

- logging – Enables logging for a ufs file system, for example:

```
# mount -o logging /dev/dsk/c0t0d0s7 /export/home
```

  UFS logging is a process of storing file system transactions or changes that make up a complete file or directory operation into a log before they are applied to the file system. After a transaction is stored, the complete transaction can be applied to the file system later.

  The UFS log is allocated from free blocks in the file system. It is sized approximately 1 Mbyte per 1 Gbyte, up to a maximum of 64 Mbytes.

As a UFS log reaches its maximum size, it begins to write transactions to the file system. When the file system is unmounted, the entire UFS log is emptied, and all transactions are written to the file system.

UFS logging offers two advantages. First, it prevents file systems from becoming inconsistent, therefore eliminating the need to run lengthy fsck scans. Secondly, you can bypass fsck scanning, which reduces the time required to reboot a system if it was stopped by a method other than an orderly shutdown.

Mounting All File Systems Manually

The /etc/vfstab file is read by the /usr/sbin/mountall command during the system boot sequence and mounts all file systems that have a yes in the mount at boot field.

The root user can use the mountall command to mount manually every file system in the /etc/vfstab file that has a yes in the mount at boot field, for example:

```
# mountall
```

To mount only the local file systems listed in the /etc/vfstab file, execute:

```
# mountall -l
```

During the boot sequence, the fsck utility checks each local file system in the /etc/vfstab file that has a device to fsck entry and an fsck pass number greater than 0. The utility determines if the file system is in a usable state to be safely mounted.

If the fsck utility determines that the file system is in an unusable state (for example, corrupted), the fsck utility repairs it before the mount is attempted. The system attempts to mount any local file systems that have a - (dash) or 0 (zero) entry in the fsck pass field without checking the file system itself.

Mounting a New File System

To add a new disk to the system, prepare the disk to hold a file system, and mount the file system, perform these general steps:

1. Set up the disk hardware, which might include setting address switches and connecting cables.
2. Perform a reconfiguration boot or run the `devfsadm` utility to add support for the new disk.
3. Use the `format` utility to partition the disk into one or more slices.
4. Create a new file system on one slice by using the `newfs` command.
5. Create a mount point for the file system by using the `mkdir` command to create a new directory in the / (root) file system.

```
# mkdir /database
```

6. Mount the new file system manually by using the `mount` command.

```
# mount /dev/dsk/c1t0d0s0 /database
```

7. Use the `mount` command to determine if the file system is mounted.

```
# mount
/ on /dev/dsk/c0t0d0s0
read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200000 on Mon
Mar 18 08:48:45 2002
/proc on /proc read/write/setuid/dev=4300000 on Mon Mar 18 08:48:44 2002
/etc/mnttab on mnttab read/write/setuid/dev=43c0000 on Mon Mar 18
08:48:44 2002
/dev/fd on fd read/write/setuid/dev=4400000 on Mon Mar 18 08:48:47 2002
/var/run on swap read/write/setuid/xattr/dev=1 on Mon Mar 18 08:48:49
2002
/tmp on swap read/write/setuid/xattr/dev=2 on Mon Mar 18 08:48:52 2002
/export/home on /dev/dsk/c0t0d0s7
read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200007 on Mon
Mar 18 08:48:52 2002
/database on /dev/dsk/c1t0d0s0
read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=800020 on Mon
Mar 18 09:15:26 2002
```

8. Edit the /etc/vfstab file to add a line entry for the new file system.

```
# vi /etc/vfstab
#device         device           mount        FS        fsck     mount    mount
#to mount       to fsck          point        type      pass     at boot  options
#
fd              -               /dev/fd       fd        -        no       -
/proc           -               /proc         proc      -        no       -
/dev/dsk/c0t0d0s1  -            -             swap      -        no       -
/dev/dsk/c0t0d0s0  /dev/rdsk/c0t0d0s0  /  ufs   1        no       -
/dev/dsk/c0t0d0s7  /dev/rdsk/c0t0d0s7  /export/home ufs  2  yes   -
swap     -        /tmp     tmpfs    -       yes     -
/dev/dsk/c1t0d0s0  /dev/rdsk/c1t0d0s0  /database  ufs  1  yes   logging
#
```

The file system automatically mounts whenever the system boots.

Mounting Different Types of File Systems

Different file system types have unique properties that affect how the mount command functions.

By default, the mount command assumes it is mounting a ufs-type file system. However, when you are mounting a different type of file system, you might have to specify its type on the command line.

You use the -F option with the mount command to specify the type of file system mounted. The file system type must be determinable from the /etc/vfstab, /etc/default/fs, or /etc/dfs/fstypes files.

Determining a File System's Type

Because the mount commands need the file system type to function properly, the file system type must be explicitly specified or determined by searching the following files:

- The /etc/vfstab file for the FS type field
- The /etc/default/fs file for a local file system type
- The /etc/dfs/fstypes file for a remote file system type

If the file system's type has not been explicitly specified on the command line using the `mount -F FStype` option, the `mount` command examines the `/etc/vfstab` file to determine the file system's type. The `mount` command makes this determination by using the file system's block device name, raw device name, or mount point directory name.

If the `mount` command cannot determine the file system's type by searching the `/etc/vfstab` file, the `mount` command uses the default file system type specified in either the `/etc/default/fs` file or the `/etc/dfs/fstypes` file, depending on whether the file system is local or remote.

The default local file system type is specified in the `/etc/default/fs` file by the line entry `LOCAL=fstype`.

```
LOCAL=ufs
```

The first line entry in the `/etc/dfs/fstypes` file determines the default remote file system type.

```
nfs NFS Utilities
autofs AUTOFS Utilities
cachefs CACHEFS Utilities
```

Using the `fstyp` Command

You can also use the `fstyp` command with the raw device name of the disk slice to determine a file system's type.

```
# fstyp /dev/rdsk/c0t0d0s7
ufs
```

Specifying an `hsfs` File System Type

To mount a file system that resides on a CD-ROM when the Volume Management services are stopped, as the `root` user perform the command:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
```

In this example, the file system type is `hsfs`, the file system resides on disk slice `/dev/dsk/c0t6d0s0`, and the mount point `/cdrom` is a pre-existing directory in the Solaris OE.

Specifying a pcfs File System Type

To mount a file system that resides on a diskette when the Volume Management services are stopped, as the root user, perform the commands:

```
# mkdir /pcfs
# mount -F pcfs /dev/diskette /pcfs
```

In this example, the file system type is pcfs. This file system resides on the device /dev/diskette, and the mount point is /pcfs.

Performing Unmounts

A file system is commonly unmounted if it needs to be checked and repaired by the `fsck` command, or if it needs to be backed up completely.

Unmounting a File System

Some file system administration tasks cannot be performed on mounted file systems.

To unmount a file system to prepare it for system maintenance, use the `umount` command.

Unmounting a file system by using the `umount` command removes it from the file system mount point and deletes its entry from the `/etc/mnttab` file.

Note – Notify users before unmounting a file system that they are currently accessing.

To unmount a file system manually by using the directory mount point, perform the command:

```
# umount /export/home
```

To unmount a file system manually by using the logical disk device name, perform the command:

```
# umount /dev/dsk/c0t0d0s7
```

Unmounting All File Systems

The `/etc/mnttab` file is read by the `/usr/sbin/umountall` command during the system `shutdown` sequence or when `umountall` is invoked from the command line. The `umountall` unmounts all file systems specified in the `vfstab` file except `/` (root), `/usr`, `/proc`, `/dev/fd`, `/var`, `/var/run`, and `/tmp`.

Run the `umountall` command as the `root` user when you want to unmount manually all the file systems listed in the `/etc/mnttab` file, for example:

```
# umountall
```

To unmount only the local file systems listed in the `/etc/mnttab` file, perform the command:

```
# umountall -l
```

To verify that a file system or a number of file systems have been unmounted, invoke the `mount` command and check the output.

Unmounting a Busy File System

Any file system that is busy is not available for unmounting. Both the `umount` and `umountall` commands display the error message:

`umount:` *`file_system_name`* `busy`

A file system is considered to be busy if one of the following conditions exists:

- A program is accessing a file or directory in the file system
- A user is accessing a directory or file in the file system
- A program has a file open in that file system
- The file is being shared

There are two methods to make a file system available for unmounting if it is busy:

- `fuser` command – Lists all of the processes that are accessing the file system and kills them if necessary
- `umount -f` command – Forces the unmount of a file system

Note – The `fuser` command displays the process IDs of all processes currently using the specified file system. Each process ID is followed by a letter code. These letter codes are described in the man page for this command.

Using the `fuser` Command

To stop all processes that are currently accessing a file system, follow these steps:

1. As the `root` user, list all of the processes that are accessing the file system. Use the following command to identify which processes need to be terminated.

```
# fuser -cu mount_point
```

This command displays the name of the file system and the user login name for each process currently active in the file system.

2. Kill all processes accessing the file system.

```
# fuser -ck mount_point
```

A `SIGKILL` message is sent to each process that is using the file system.

3. Verify that there are no processes accessing the file system.

```
# fuser -c mount_point
```

4. Unmount the file system.

```
# umount mount_point
```

Using the `umount -f` Command

As the `root` user, you can unmount a file system, even if it is busy, by using the `-f` (force) option with the `umount` command. The following is the format for this command:

```
# umount -f mount_point
```

The file system is unmounted even if it contains open files. A forced unmount can result in loss of data and in zombie processes that are left running on the system. However, it is particularly useful for unmounting a shared file system if the remote file server is nonfunctional.

Repairing Important Files if Boot Fails

The following procedure describes how to boot from the Solaris OE Software CD-ROM to edit a misconfigured `/etc/vfstab` file.

1. Insert the Solaris 9 OE Software 1 of 2 CD-ROM into the CD-ROM drive.
2. Execute a single-user boot from the CD-ROM.

```
ok boot cdrom -s
Boot device: /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f File and args -s
SunOS Release 5.9 Generic 64 bit
Copyright 1983-2002 by Sun Microsystems, Inc. All rights reserved.
Configuring /dev and /devices
Use is subject to license terms
Using RPC Bootparams for network configuration information.
Skipping interface hme0
-
INIT: SINGLE USER MODE
#
```

Note – Performing a single-user boot operation from this software CD-ROM creates an *in-memory* copy of the `/var` file system, which supports your ability to perform administrative tasks.

3. Use the `fsck` command on the `/` (root) partition to check and repair any potential problems in the file system.

```
# fsck /dev/rdsk/c0t0d0s0
```

4. If the `fsck` command is successful, mount the `/` (root) file system on the `/a` directory to gain access to the file system on disk.

```
# mount /dev/dsk/c0t0d0s0 /a
```

5. Set and export the `TERM` variable, which enables the `vi` editor to work properly.

```
# TERM=sun
# export TERM
```

6. Edit the `/etc/vfstab` file, and correct any problems. Then exit the file.

```
# vi /a/etc/vfstab
:wq!
```

7. Unmount the file system.

```
# cd /
# umount /a
```

8. Reboot the system.

```
# init 6
```

Accessing Mounted Diskettes and CD-ROMs

To provide access to file systems on diskettes and CD-ROMs, the Solaris OE provides users a standard interface referred to as Volume Management.

Note – The Solaris 9 OE includes support for additional removable media such as DVDs, Jaz drives, Zip drives, and PCMCIA memory cards. (PCMCIA stands for Personal Computer Memory Card International Association.) For more information on using these devices, see the resources available on the Solaris 9 Documentation CD.

Volume Management provides two major benefits:

- It automatically mounts removable media for both the `root` user and non-`root` users.
- It can give other systems on the network automatic access to any removable media currently inserted in the local system.

The Volume Management service is controlled by the `/usr/sbin/vold` daemon. By default, this service is always running on the system so that it can automatically manage diskettes and CD-ROMs for regular users.

Volume Management features automatic detection of CD-ROMs. However, it does not detect the presence of a diskette that has been inserted in the drive until the `volcheck` command is run. This command instructs the `vold` daemon to check the diskette drive for any inserted media. Volume Management can mount `ufs`, `pcfs`, `hsfs`, and `udfs` file systems.

Using Volume Management

To make working with diskettes and CD-ROMs simple for your users, each device is easy to mount and mounts at an easy-to-remember location.

If the `vold` daemon detects that the mounted device contains a file system, then the device is mounted at the directory location.

Table 4-1 lists the directory locations of mounted devices that contain file systems.

Table 4-1 Directory Locations

Media Device	Access File Systems On
First diskette drive	`/floppy/floppy0`
First CD-ROM or DVD drive	`/cdrom/cdrom0`
First Jaz drive	`/rmdisk/jaz0`
First Zip drive	`/rmdrive/zip0`
First PCMCIA card	`/pcmem0`

If the `vold` daemon detects that the mounted device does not contain a file system, the device is accessible through a path.

Table 4-2 lists the paths for mounted devices that do not contain file systems.

Table 4-2 Paths for Accessing Devices

Media Device	Access Raw Device On
First diskette drive	`/vol/dev/aliases/floppy0`
First CD-ROM or DVD drive	`/vol/dev/aliases/cdrom0`
First Jaz drive	`/vol/dev/aliases/jaz0`
First Zip drive	`/vol/dev/aliases/zip0`
First PCMCIA card	`/vol/dev/aliases/pcmem0`

When Volume Management is running on the system, a regular user can easily access a diskette or CD-ROM by following these basic steps:

1. Insert the media.
2. For diskettes only, enter the `volcheck` command.
3. Use the `cd` command to change to the directory of the mounted volume.
4. Work with files on the media.
5. Use the `cd` command to leave the directory structure of the mounted volume.
6. Eject the media.

Table 4-3 shows the configuration files used by Volume Management.

Table 4-3 Volume Management Configuration Files

File	Description
`/etc/vold.conf`	The Volume Management configuration file. This file defines items, such as what action should be taken when media is inserted or ejected, which devices are managed by Volume Management, and which file system types are unsafe to eject.
`/etc/rmmount.conf`	The `rmmount` command configuration file. The `rmmount` command is a removable media mounter that is executed by the Volume Management daemon whenever a CD-ROM or diskette is inserted.

Restricting Access to Mounted Diskettes and CD-ROMs

To restrict regular users from accessing diskettes or CD-ROMs on the system, you can, as the `root` user, terminate the Volume Management service.

Stopping Volume Management

To stop Volume Management from running on a system temporarily, as the `root` user perform the command:

```
# /etc/init.d/volmgt stop
```

To restart the Volume Management service, as the `root` user perform the command:

```
# /etc/init.d/volmgt start
```

Troubleshooting Volume Management Problems

If a CD-ROM fails to eject from the drive, as the `root` user attempt to stop Volume Management. If this is unsuccessful, kill the `vold` daemon.

```
# /etc/init.d/volmgt stop
```

or as a last resort:

```
# pkill -9 vold
```

Push the button on the system to eject the CD-ROM. The CD-ROM tray ejects. Remove the CD-ROM, and leave the tray out. Then restart the Volume Management service.

```
# /etc/init.d/volmgt start
```

Wait a few seconds, and then push the CD-ROM tray back into the drive.

Accessing a Diskette or CD-ROM Without Volume Management

When Volume Management is not running, only the `root` user can mount and access a diskette or CD-ROM. Follow these steps:

1. Insert the media device.
2. Become the `root` user.
3. Create a mount point, if necessary.
4. Determine the file system type.
5. Mount the device by using the mount options listed in the following sections.
6. Work with files on the media device.
7. Unmount the media device.
8. Eject the media device.
9. Exit the `root` session.

Using the `mount` Command

To mount a file system that resides on a CD-ROM when the Volume Management services are stopped, as the `root` user, perform the command:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
```

In this example, the file system type is `hsfs`, the file system resides on disk slice `/dev/dsk/c0t6d0s0`, and the mount point `/cdrom` is a pre-existing directory in the Solaris OE.

To mount a file system that resides on a diskette when the Volume Management services are stopped, as the `root` user, perform the command:

```
# mkdir /pcfs
# mount -F pcfs /dev/diskette /pcfs
```

In this example, the file system type is `pcfs`. This file system resides on the `/dev/diskette` device, and the mount point used is `/pcfs`.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Mounting File Systems (Level 1)

In this exercise, you complete the following tasks:

- Create mount points
- Mount file systems
- Specify mount options

Preparation

This exercise requires a spare disk that contains four unmounted `ufs` file systems on Slices 0, 1, 3, and 4. Refer to the lecture notes as necessary to perform the tasks listed.

Tasks

Complete the following tasks:

- Record the default mount options that are used by the `/` (root) file system mounted on your system. Mount the file system found on Slice 4 of your spare disk on the `/morespace` directory. Verify the mount options applied to the `/morespace` file system.

  (Steps 1–3 in the Level 2 lab)
- Create a new file in the `/morespace` file system that contains one line of text. Record the modify time for this file. Use the `ls` command to display the last access time for this file. Record the time value. Wait one minute, and then display the file content. Again check and record the last access time for this file.

  (Steps 4–7 in the Level 2 lab)
- Unmount the `/morespace` file system. Remount the same file system as `/morespace`, and use the `noatime` mount option. Again display the content of your text file. Check and record the last access time for it. Add a line to the `/etc/vfstab` file that mounts the `/morespace` file system when the system reboots. Reboot the system by using the `reboot` command, and verify that the `/morespace` file system is mounted.

  (Steps 8–11 in the Level 2 lab)

- Mount the file system on Slice 0 as `/dir0`. Mount the file system on Slice 1 as `/dir0/dir1`. In a second terminal window, change to the `/dir0/dir1` directory. In the original terminal window, try to unmount the `/dir0/dir1` directory. Record the error messages. Attempt to forcibly unmount the `/dir0/dir1` directory. Record the result. Attempt to use the `pwd` command in the second terminal window. Record what happens.

  (Steps 12–17 in the Level 2 lab)

Exercise: Mounting File Systems (Level 2)

In this exercise, you complete the following tasks:

- Create mount points
- Mount file systems
- Specify mount options

Preparation

This exercise requires a spare disk that contains four unmounted `ufs` file systems on Slices 0, 1, 3, and 4. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Record the default mount options that are used by the / (root) file system mounted on your system. Mount the file system found on Slice 4 of your spare disk on the `/morespace` directory. Verify the mount options applied to the `/morespace` file system.
- Create a new file in the `/morespace` file system that contains one line of text. Record the modify time for this file. Use the `ls` command to display the last access time for this file. Record the time value. Wait one minute, and then display the file content. Again check and record the last access time for this file.
- Unmount the `/morespace` file system. Remount the same file system as `/morespace`, and use the `noatime` mount option. Again display the content of your text file. Check and record the last access time for it. Add a line to the `/etc/vfstab` file that mounts the `/morespace` file system when the system reboots. Reboot the system using the `reboot` command, and verify that the `/morespace` file system is mounted.
- Mount the file system on Slice 0 as `/dir0`. Mount the file system on Slice 1 as `/dir0/dir1`. In a second terminal window, change to the `/dir0/dir1` directory. In the original terminal window, try to unmount the `/dir0/dir1` directory. Record the error messages. Attempt to forcibly unmount the `/dir0/dir1` directory. Record the result. Attempt to use the `pwd` command in the second terminal window. Record what happens.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Use the `mount` command to list the file systems that are currently mounted on your system. What are the default mount options applied to the / (root) file system?
2. Create the directory `/morespace` to use as the mount point.
3. Mount the file system on Slice 4 of your spare disk to the `/morespace` directory. Record the default mount options that were applied to this mount.
4. Change to the `/morespace` directory, and create a new file that has one line of content.
5. Display a long listing for this file, and record the time value it reports. This time value represents when the file was last modified.
6. Add the `-u` option to the `ls` command to show when the file was last accessed. This time value is updated whenever you read the file.
7. Wait one minute or more, and then use the `cat` command to display the file. Again check and record the access time. The access time should differ from the access time indicated in the previous step.
8. Change to the / (root) directory. Unmount the `/morespace` file system. Remount the same file system to the `/morespace` directory, but add the option that prevents update of access time values. Verify that the options to the mount were applied.
9. Return to the `/morespace` file system, and use the `cat` command to display your test file. Again check and record the access time. The access time should match the access time that existed prior to your unmounting and mounting the `/morespace` file system.
10. Add a line to the `/etc/vfstab` file to make the mount for the `/morespace` file system happen when you boot the system.
11. Reboot your system. Log in as the `root` user, and open a terminal window. Use the `mount` command to verify that the `/morespace` file system is mounted.
12. Create a directory called `/dir0`. Mount the file system that resides on Slice 0 of your spare disk as `/dir0`.
13. Create a directory called `/dir0/dir1`. Mount the file system that resides on Slice 1 of your spare disk as `/dir0/dir1`.

14. Open a second terminal window. In this new window, change the directory to /dir0/dir1.
15. In your original terminal window, attempt to unmount the file system mounted below the /dir0/dir1 directory. Which message is displayed? Does the file system unmount?

Note – To discover why you could not unmount the file system, use the `fuser -cu /dir0/dir1` command. The `fuser` command should show the process ID of the shell.

16. In your original terminal window, again attempt to unmount the file system mounted below the /dir0/dir1 directory. Add the `-f` option to the `umount` command. Which message is displayed? Does the file system unmount?
17. In the second terminal window, attempt to determine your current working directory. Which message is displayed? Change the directory to / (root), and verify that the `pwd` command works.

Tasks and Solutions

Complete the following steps:

1. Log in as the `root` user, and open a terminal window. Use the `mount` command to list the file systems that are currently mounted on your system. What are the default mount options applied to the / (root) file system?

```
# mount
read/write/setuid/intr/largefiles/xattr/onerror=panic/
dev=2200000
```

Your dev= *number depends on the architecture of your system.*

2. Create the directory /morespace to use as the mount point.

```
# mkdir /morespace
```

3. Mount the file system on Slice 4 of your spare disk to the /morespace directory. Record the default mount options that were applied to this mount.

```
# mount /dev/dsk/c1t0d0s4 /morespace
# mount
read/write/setuid/intr/largefiles/xattr/onerror=panic/
dev=80001c
```

4. Change to the /morespace directory, and create a new file that has one line of content.

```
# cd /morespace
# cat > testfile
This is a test.
<crtl> d
#
```

5. Display a long listing for this file, and record the time value it reports. This time value represents when the file was last modified.

```
# ls -l
```

6. Add the `-u` option to the `ls` command to show when the file was last accessed. This time value is updated whenever you read the file.

```
# ls -lu
```

7. Wait one minute or more, and then use the `cat` command to display the file. Again check and record the access time. The access time should differ from the access time indicated in the previous step.

```
# cat testfile
This is a test
# ls -lu
```

Exercise: Mounting File Systems (Level 3)

In this exercise, you complete the following tasks:

- Create mount points
- Mount file systems
- Specify mount options

Preparation

This exercise requires a spare disk that contains four unmounted `ufs` file systems on Slices 0, 1, 3, and 4. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Record the default mount options that are used by the / (root) file system mounted on your system. Mount the file system found on Slice 4 of your spare disk on the `/morespace` directory. Verify the mount options applied to the `/morespace` file system.
- Create a new file in the `/morespace` file system that contains one line of text. Record the modify time for this file. Use the `ls` command to display the last access time for this file. Record the time value. Wait one minute, and then display the file content. Again check and record the last access time for this file.
- Unmount the `/morespace` file system. Remount the same file system as `/morespace`, and use the `noatime` mount option. Again display the content of your text file. Check and record the last access time for it. Add a line to the `/etc/vfstab` file that mounts the `/morespace` file system when the system reboots. Reboot the system using the `reboot` command, and verify that the `/morespace` file system is mounted.
- Mount the file system on Slice 0 as `/dir0`. Mount the file system on Slice 1 as `/dir0/dir1`. In a second terminal window, change to the `/dir0/dir1` directory. In the original terminal window, try to unmount the `/dir0/dir1` directory. Record the error messages. Attempt to forcibly unmount the `/dir0/dir1` directory. Record the result. Attempt to use the `pwd` command in the second terminal window. Record what happens.

8. Change to the / (root) directory. Unmount the `/morespace` file system. Remount the same file system to the `/morespace` directory, but add the option that prevents update of access time values. Verify that the options to the mount were applied.

```
# cd /
# umount /morespace
# mount -o noatime /dev/dsk/c1t0d0s4 /morespace
# mount
```

9. Return to the `/morespace` file system, and use the `cat` command to display your test file. Again check and record the access time. The access time should match the access time that existed prior to your unmounting and mounting the `/morespace` file system.

```
# cd /morespace
# cat testfile
This is a test
# ls -lu
```

10. Add a line to the `/etc/vfstab` file to make the mount for the `/morespace` file system happen when you boot the system.

```
/dev/dsk/c1t0d0s4   /dev/rdsk/c1t0d0s4   /morespace  ufs  2   yes  noatime
```

11. Reboot your system. Log in as the `root` user, and open a terminal window. Use the `mount` command to verify that the `/morespace` file system is mounted.

```
# init 6
(reboot messages & login prompts)
# mount
```

12. Create a directory called `/dir0`. Mount the file system that resides on Slice 0 of your spare disk as `/dir0`.

```
# mkdir /dir0
# mount /dev/dsk/c1t0d0s0 /dir0
```

13. Create a directory called `/dir0/dir1`. Mount the file system that resides on Slice 1 of your spare disk as `/dir0/dir1`.

```
# mkdir /dir0/dir1
# mount /dev/dsk/c1t0d0s1 /dir0/dir1
```

14. Open a second terminal window. In this new window, change the directory to `/dir0/dir1`.

```
# cd /dir0/dir1
```

15. In your original terminal window, attempt to unmount the file system mounted below the `/dir0/dir1` directory. What message is displayed? Does the file system unmount?

```
# umount /dev/dsk/c1t0d0s1
# mount
umount: /dir0/dir1 busy
```

The file system does not unmount.

Note – To discover why you could not unmount the file system, use the `fuser -cu /dir0/dir1` command. The `fuser` command should show the process ID of the shell.

---

16. In your original terminal window, again attempt to unmount the file system mounted below the `/dir0/dir1` directory. Add the `-f` option to the `umount` command. Which message is displayed? Does the file system unmount?

```
# umount -f /dev/dsk/c1t0d0s1
# mount
```

No messages are displayed. The file system unmounts.

17. In the second terminal window, attempt to determine your current working directory. Which message is displayed? Change the directory to `/` (root), and verify that the `pwd` command works.

```
# pwd
Cannot determine current directory.
# cd /
# pwd
/
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 5

Installing the Solaris™ 9 Operating Environment

Objectives

Upon completion of this module, you should be able to:

- Identify the fundamentals of CD-ROM installation
- Install the Solaris™ 9 Operating Environment (Solaris 9 OE) from a CD-ROM

The following course map shows how this module fits into the current instructional goal.

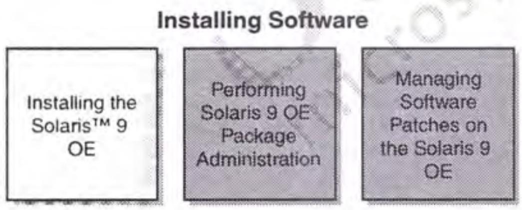

Figure 5-1 Course Map

Identifying the Fundamentals of the CD-ROM Installation

The following section describes the CD-ROM installation of the Solaris 9 OE.

Solaris 9 OE Installation and Upgrade Options

There are a number of ways to install the Solaris 9 OE on your system. They include:

- Solaris `suninstall` program
- Solaris Web Start Installation software
- Custom JumpStart™ procedure
- Solaris Web Start Flash installation
- Standard upgrade
- Solaris Live Upgrade method

This module focuses on the `suninstall` program.

Solaris Web Start 3.0 Installation Software

The Solaris Web Start 3.0 Installation is a software application powered by Java™ technology. The Solaris Web Start program uses a graphical user interface (GUI). The GUI guides you through the installation of the Solaris 9 OE and other software on a single system from either a local or remote CD-ROM drive. Boot the system from the Solaris 9 Installation CD-ROM to invoke the Solaris Web Start program.

The Solaris Web Start program enables you to move easily forward and backward during the installation process to make changes as needed. Installation tasks are divided into panels that prompt you to enter system configuration information. The panels also offer you the opportunity to select default values.

You must interact with Solaris Web Start throughout the installation or upgrade procedure. Consequently, this method might not be the most efficient method when you need to install or upgrade several systems.

Custom JumpStart™ Installation

The Solaris JumpStart procedure installs Solaris OE software on a system by referencing a user-defined profile. You can customize profiles for different types of systems.

A JumpStart installation provides a hands-off installation across the network and is based on a central-configured server. The JumpStart procedure is a command-line interface that enables you to incorporate shell scripts. The shell scripts include pre-installation and post-installation tasks. You choose the profile and the scripts to use for installation or upgrade. Then the custom installation method installs or upgrades the system.

Solaris Web Start Flash Installation Software

The Solaris Web Start Flash Installation software enables you to install many systems based on a configuration that you install on a master system. After you have installed and configured the master system, you create a flash archive from the master system. You create as many flash archives as you need and choose which flash archive to install on each system.

The standard Solaris OE installation methods install each Solaris OE package individually. This method of package-based installation is time consuming because the installation must update the package map for each package. The Solaris Web Start Flash software's archive installs on your system much faster than when you install each of the individual Solaris OE packages, because you are only producing a copy of an already installed system.

Solaris upgrade options include both the standard upgrade and the live upgrade.

Standard Upgrade to the Solaris OE

A standard upgrade merges the new version of the Solaris OE with the existing files on the system's disk. The methods available for a standard upgrade are Solaris `suninstall` program, the Solaris Web Start software, and the custom JumpStart procedure.

A standard upgrade saves many of the modifications that were made to the OE with the previous version of the Solaris OE. Because the Solaris OE is unavailable to users during the standard upgrade, the standard upgrade results in longer periods of downtime.

Solaris Live Upgrade Software

The Solaris Live Upgrade Software upgrades a duplicate boot environment while the active boot environment is still running. This method eliminates downtime of the production environment. The Solaris Live Upgrade method can be run with either a GUI or a command-line interface. First, create a duplicate boot environment. After that has been created, upgrade or install a Solaris Web Start Flash archive on the inactive boot environment. When you are ready, activate the inactive boot environment. During the next reboot, the inactive boot environment becomes the active boot environment. If there is a failure, you can recover your original boot environment by reactivating it and rebooting the system.

Solaris Live Upgrade Software requires enough available disk space to create a duplicate of your boot environment. To estimate the file system size needed to create a boot environment, start the creation of the new boot environment. The file system size is calculated, and you can then abort the process.

Hardware Requirements for Installation of the Solaris 9 OE

A Solaris 9 OE installation requires:

- A system based on an UltraSPARC® processor
- 64 Mbytes of memory
- 2.3 Gbytes of disk space
- A keyboard and monitor
- Access to a CD-ROM drive or an installation server

Software Components of the Solaris OE

The Solaris OE software is organized into three components:

- Software packages
- Software clusters
- Software groups

Figure 5-2 shows the relationship among the Common Desktop Environment (CDE) software components.

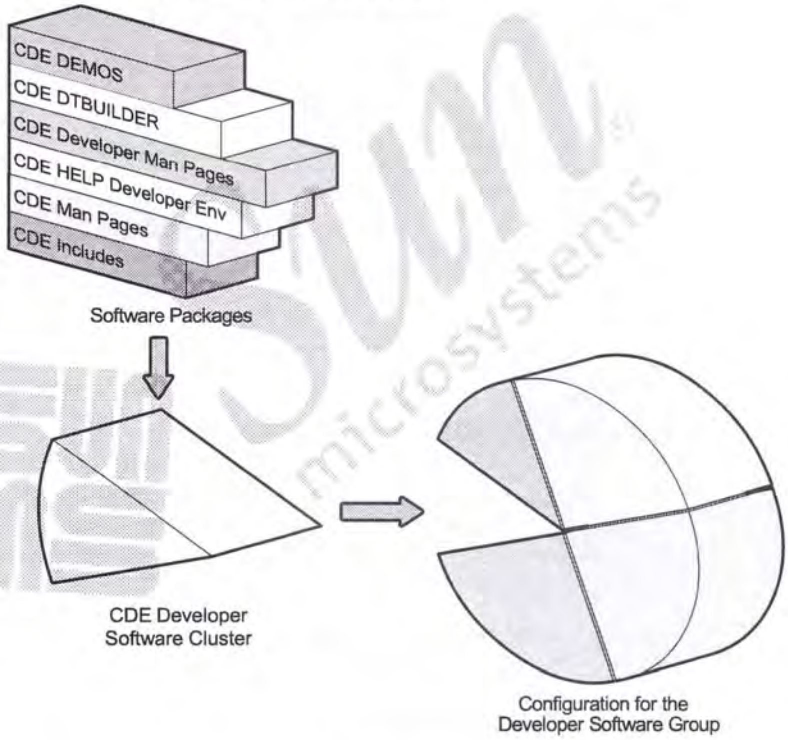

Figure 5-2 Solaris OE Software Components

Software Packages

A software package contains a group of software files and directories. The package also contains the related software installation scripts.

Software Clusters

During the software installation process, software clusters group logical collections of software packages together. Table 5-1 shows the software packages that are grouped into the CDE software cluster.

Table 5-1 Packages Included in the CDE Software Cluster

SUNWdtab	SUNWdthed	SUNWdtmad	SUNWeudhr
SUNWdtbas	SUNWdthev	SUNWdtrme	SUNWeudhs
SUNWdtdem	SUNWdticn	SUNWdtwn	SUNWeudis
SUNWdtdmn	SUNWdtim	SUNWeudba	SUNWeudlg
SUNWdtdst	SUNWdtinc	SUNWeudbd	SUNWmfman
SUNWdthe	SUNWdtma	SUNWeudda	

Some software clusters contain only one software package.

Solaris OE Software Groups

Software groups are collections of Solaris OE software packages. Each software group includes support for different functions and hardware drivers. The Solaris OE is made up of five software groups:

- Core
- End User System Support
- Developer System Support
- Entire Distribution
- Entire Distribution Plus Original Equipment Manufacturers (OEM)

Figure 5-3 shows the software groups that compose the Solaris OE.

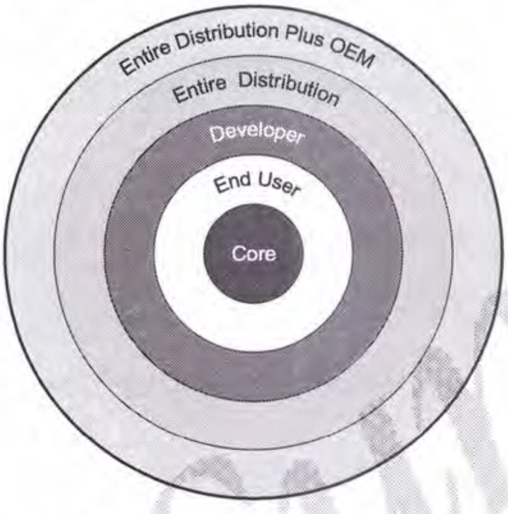

Figure 5-3 Solaris OE Software Groups

Core Software Group (`SUNWCreq`)

The Core software group contains the minimum software required to boot and run the Solaris OE in a minimum configuration, without the support to run many server applications. The Core software group includes a minimum of networking software, including Telnet, File Transfer Protocol (FTP), Network File System (NFS), Network Information Service (NIS) clients, and Domain Name Service (DNS). This software group also includes the drivers required to run the Common Desktop Environment (CDE) but does not include the CDE software. The Core software group also does not include online manual pages.

End User System Support Software Group (`SUNWCuser`)

The End User System Support software group contains the Core software group and also contains the recommended software for an end user plus the CDE.

Developer System Support Software Group (`SUNWCprog`)

The Developer System Support software group contains the End User System Support software group. It also contains the libraries, the include files, the online manual pages, and the programming tools for developing software.

Entire Distribution Software Group (`SUNWCall`)

The Entire Distribution software group contains the Developer System Support software group. It also contains additional software needed for servers. The software that is in the Entire Distribution software group is the entire Solaris OE software release minus OEM support.

Entire Distribution Plus OEM Support Software Group (`SUNWCXall`)

The Entire Distribution Plus OEM Support software group contains the entire Solaris OE software release. It also contains additional hardware support for OEMs. This software group is recommended when you are installing the Solaris OE software on non-Sun servers that use UltraSPARC processors.

To view the names of the cluster configurations, perform the command:

```
# grep METACLUSTER /var/sadm/system/admin/.clustertoc
METACLUSTER=SUNWCXall
METACLUSTER=SUNWCall
METACLUSTER=SUNWCprog
METACLUSTER=SUNWCuser
METACLUSTER=SUNWCreq
METACLUSTER=SUNWCmreq
```

Note – The metacluster `SUNWCmreq` is a hidden metacluster. It allows you to create a minimal core metacluster by de-selecting packages from the core metacluster.

To determine which cluster configuration has been installed on the system, perform the command:

```
# cat /var/sadm/system/admin/CLUSTER
CLUSTER=SUNWCXall
```

Installing the Solaris 9 OE From a CD-ROM

In the Solaris 9 OE, the `suninstall` program uses Tab Window Manager (TWM) for window management.

TWM provides the following features:

- Title bars
- Shaped windows
- Several forms of icon management
- User-defined macro functions
- A click-to-type and pointer-driven keyboard focus
- User-specified key and pointer button bindings

Pre-Installation Information

Consider the following general guidelines while planning an installation:

- Allocate additional disk space for each language that you install.
- Allocate additional space in the `/var` file system if you plan to have your system support printing or mail.
- Allocate double the amount of physical memory in the `/var` file system if you plan to use the crash dump feature `savecore` on your system.
- Allocate additional space in the `/export` file system if you plan to provide a home directory file system for users on the Solaris 9 OE.
- Allocate space for the Solaris software group you want to install.
- Allocate an additional 30 percent more disk space for each file system that you create, and create a minimum number of file systems. This leaves room for upgrades to future software releases.

Note – By default, the Solaris OE installation methods create only the `/` (root) file system and the `swap` partition.

- Allocate additional disk space for additional software or third-party software.

Before installing the Solaris OE software on a networked standalone system, you must provide the following information:

Host name	Determine a unique, and usually, short name for the networked system. You can use the command `uname -n` to find the host name on an existing system.
Host Internet Protocol (IP) address	Determine the software address that represents the host address and network address. You can use the `ifconfig` *`interface`* command (for example, `ifconfig hme0`) to display your current IP address.
Name service type	Determine if the networked system is to be included in one of the following types of name service domains: Lightweight Directory Access Protocol (LDAP), NIS, Network Information Service Plus (NIS+), DNS, or none.
Subnet mask	Determine if the networked system is included in a particular subnet. The subnet mask is stored in the `/etc/netmasks` file.

Note – A subnet is used to partition network traffic. Segmenting network traffic over many different subnets increases the bandwidth available to each host.

Geographic location and time zone	Determine the specific region where the networked system physically resides.
`root` password	Determine a password assigned to the `root` user. Use the `root` password to gain access to `root` privileges on the networked system.
Language	Determine the language with which to install the Solaris OE. Use the CD-ROM labeled Solaris 9 Installation SPARC® Platform Edition. The installation software enables the user to choose from a list of languages. Prompts, messages, and other installation information are displayed in the chosen language. The language choices include English, German, Spanish, French, Italian, Japanese, Korean, Swedish, Simplified Chinese, and Traditional Chinese.

As the last step in the pre-installation process, make sure the following Solaris 9 OE CD-ROM set is available:

- Solaris 9 Installation English SPARC Platform Edition or Solaris 9 Installation Multilingual SPARC Platform Edition
- Solaris 9 Software 1 of 2 CD-ROM SPARC Platform Edition and Solaris 9 Software 2 of 2 CD-ROM SPARC Platform Edition
- Solaris 9 Languages SPARC Platform Edition

Before performing a software installation, always back up any modifications or data that exist in the previous version of the Solaris OE, and restore them after completing the installation.

Demonstration: Performing an Interactive Installation

In this demonstration, your instructor leads you through an interactive installation of the Solaris 9 OE.

Preparation

The interactive installation demonstration requires a networked, standalone system configured with a 2-Gbyte, or larger, boot disk. Depending on the speed of CD-ROM devices in use, the complete installation process requires approximately one hour.

Demonstration Summary

Boot the system from the Solaris 9 CD-ROM 1 of 2, and install the Solaris 9 OE software. Create a configuration as follows:

- Assign host name, IP address, subnet mask, routing, time zone, and naming service information compatible with the classroom configuration.
- Perform an initial installation, and use the Entire Distribution configuration cluster.
- Select the appropriate boot disk, and manually lay out your file system.
- Create slices for the `/` (root), `swap`, `/opt`, `/usr`, and `/export/home` file systems.
- Elect not to install additional software products.
- Set the `root` password to `cangetin`.

Demonstration Instructions

Complete the following steps:

1. Insert the Solaris 9 OE CD-ROM 1 of 2 into the CD-ROM drive. If the current version of the Solaris OE is running, log in as `root`, and bring the system to run level 0.

```
# init 0
```

   You can also abort the Solaris OE by pressing the Stop-A key sequence.

2. Boot the system from the CD-ROM. Ignore error messages, such as cable problem messages, that relate to network interfaces that are not attached.

```
ok boot cdrom
```

   The installation program is loaded into random access memory (RAM), and the installation process begins automatically. However, no changes are made to the disk until you click the Begin Installation button at the end of the installation process.

3. When the installation software is finished loading, a list of languages appears. Prompts, messages, and other installation information are displayed in the chosen language. English is the default language choice. You can select a different language from the list of available languages.

```
 Select a Language

0   English
1   French
2   German
3   Italian
4   Japanese
5   Korean
6   Simplified Chinese
7   Spanish
8   Swedish
9   Traditional Chinese


Please make a choice (0-9), or press h or ? for help:
```

   Respond by making your language selection.

4. A list of locales appears. Respond by making your locale selection.

```
   Select a Locale

58 U.S.A. (en_US.ISO8859-15)

   ... < remaining lines removed > ...
```

If the Solaris `suninstall` program detects a frame buffer for the system, it uses the GUI. If it does not detect a frame buffer, the `suninstall` program uses the command-line interface. The content and sequence of instructions in both are generally the same. This demonstration uses the GUI.

The system responds with:

```
Starting Window System
```

The TWM starts up, and two windows appear. Figure 5-4 shows the first of these windows to appear. The Solaris Install Console window remains open throughout the installation and provides information pertinent to the installation.

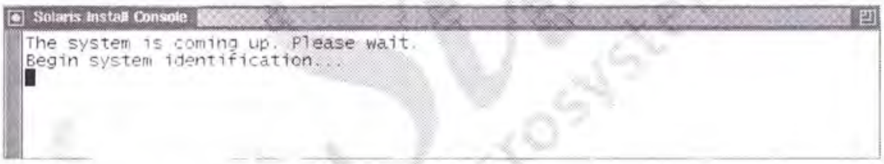

Figure 5-4 Solaris Install Console Window

Figure 5-5 shows the second window that appears when the TWM starts. The Solaris Installation Program window provides a brief description of the installation process.

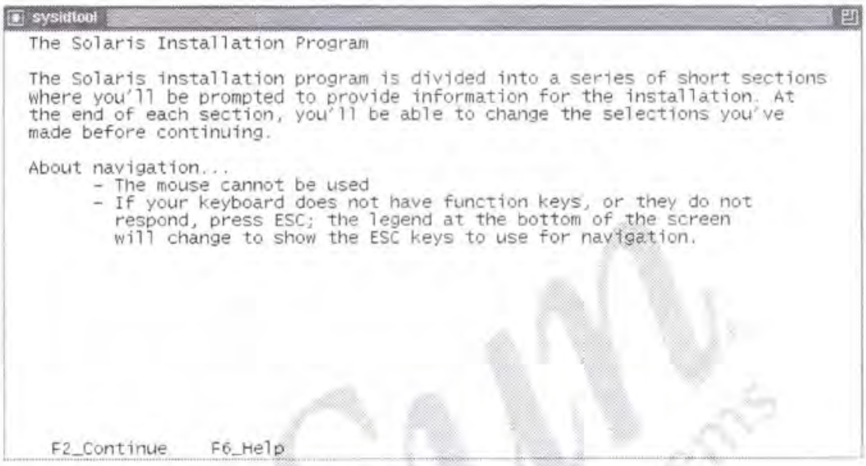

Figure 5-5 Solaris Installation Program Window

5. When you have read the brief description, press F2.

   Figure 5-6 shows the next window that appears.

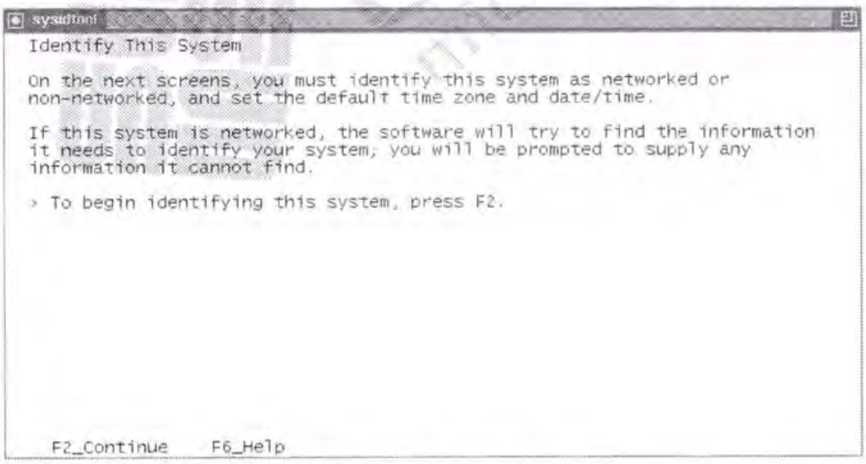

Figure 5-6 Identify This System Window

6. Read the description of the identification process. When you are finished, press F2.

   Figure 5-7 shows the next window that appears.

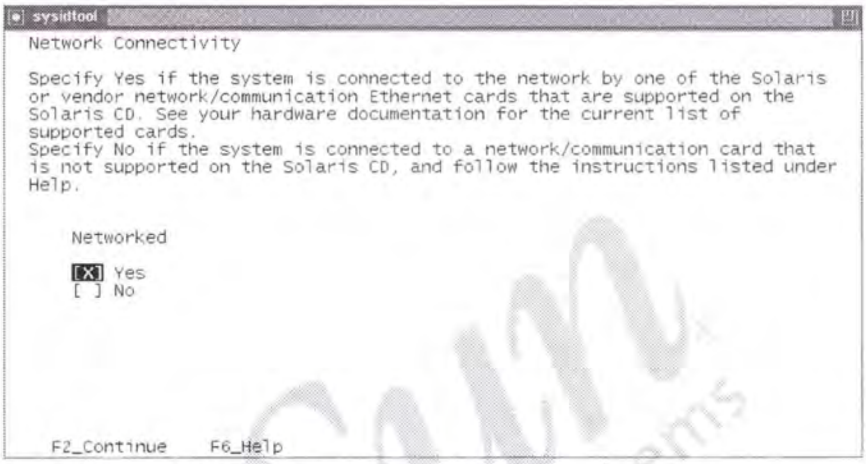

Figure 5-7 Network Connectivity Window

7. Select `Yes` if your system is connected to a network. Use the arrow keys to move between choices, and press Return to select the choice. Use these steps to make selections for the remainder of the demonstration. When you have made your selection, press F2.

Figure 5-8 shows the next window that appears.

```
sysidtool
DHCP

On this screen you must specify whether or not this system should use DHCP
for network interface configuration.  Choose Yes if DHCP is to be used, or
No if the interfaces are to be configured manually.

NOTE: DHCP support will not be enabled, if selected, until after the system
reboots.

    Use DHCP

    [ ] Yes
    [X] No




  F2_Continue    F6_Help
```

Figure 5-8 DHCP Choice Window

8. Select `No` to confirm that the system is not using Dynamic Host Configuration Protocol (DHCP) for network interface information. To continue, press F2.

9. If your system has more than one network interface, Figure 5-9 appears.

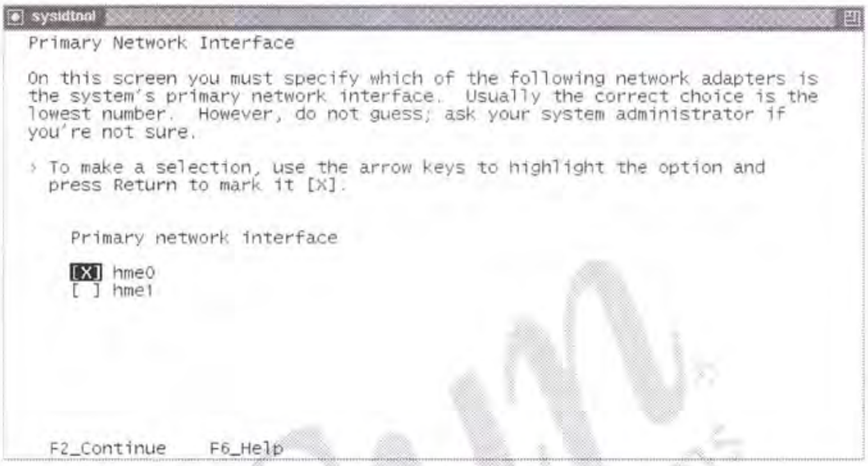

Figure 5-9 Primary Network Interface Window

Select which network interface you want to make your primary interface (in this example, choose `hme0`). Press F2 to continue.

Figure 5-10 shows the next window that appears.

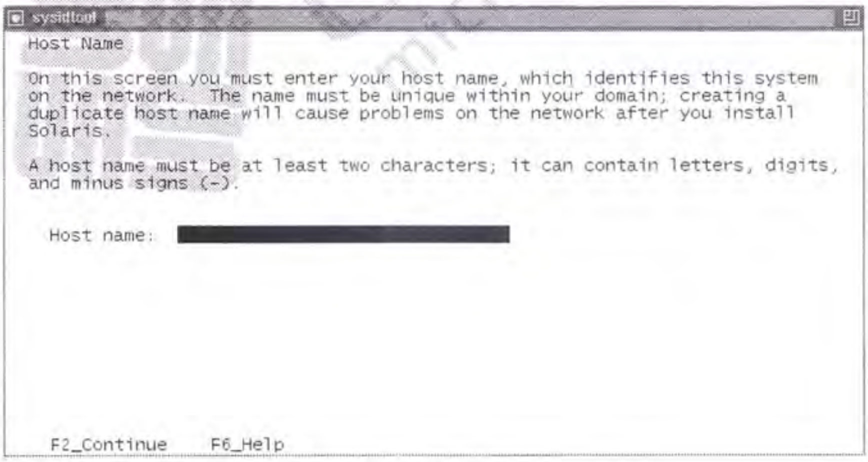

Figure 5-10 Host Name Window

10. Enter the assigned host name for the system in the Host name field. To continue, press F2.

    Figure 5-11 shows the next window that appears.

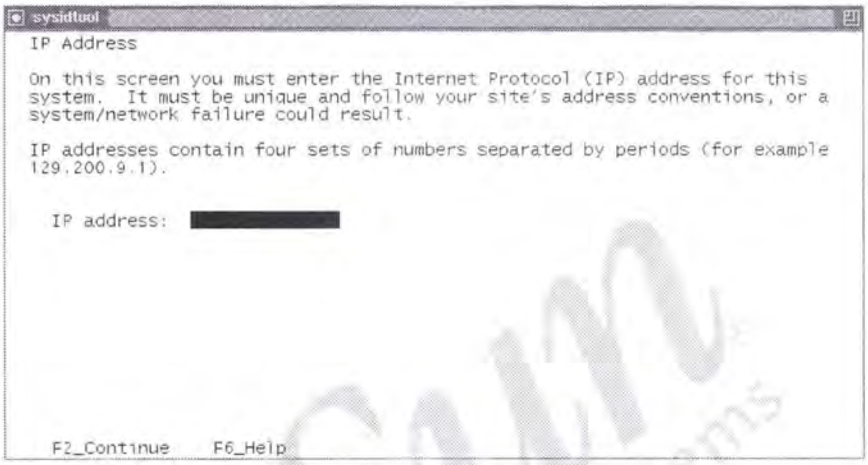

Figure 5-11 IP Address Window

11. Enter the assigned IP address in the IP address field. To continue, press F2.

    Figure 5-12 shows the next window that appears.

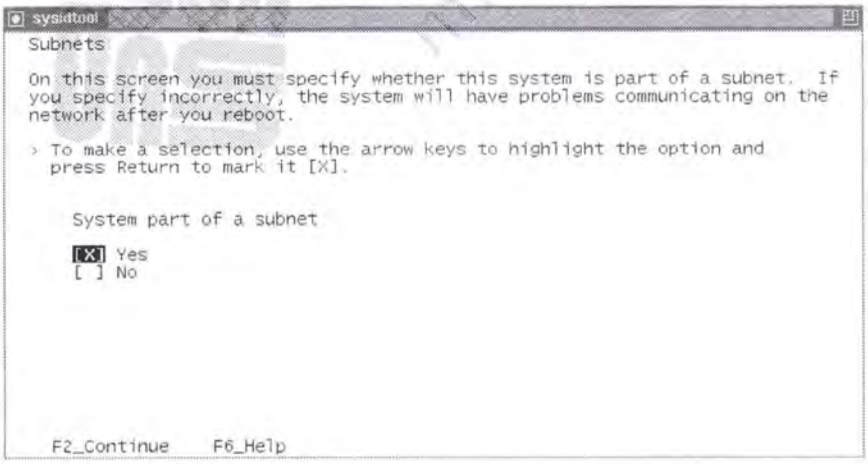

Figure 5-12 Subnets Window

12. Select `Yes` to confirm that the system is part of a subnet. To continue, press F2.

    Figure 5-13 shows the next window that appears.

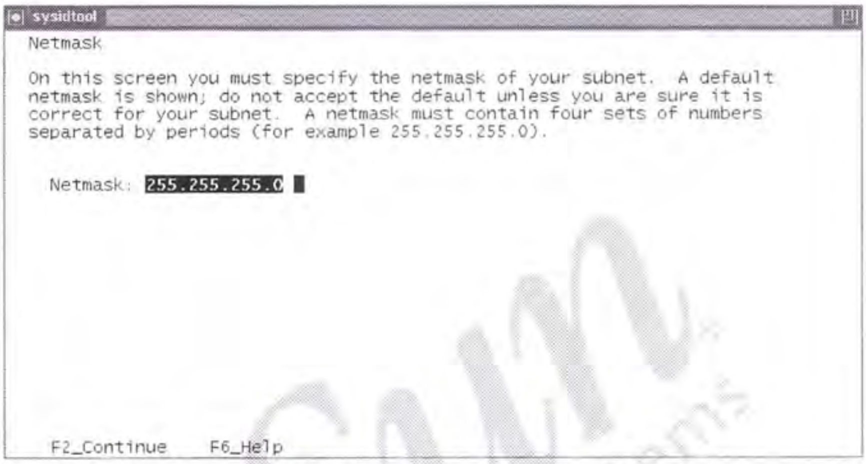

Figure 5-13 Netmask Window

13. For this demonstration, accept the default subnet mask of `255.255.255.0`. To continue, press F2.

    Figure 5-14 shows the next window that appears.

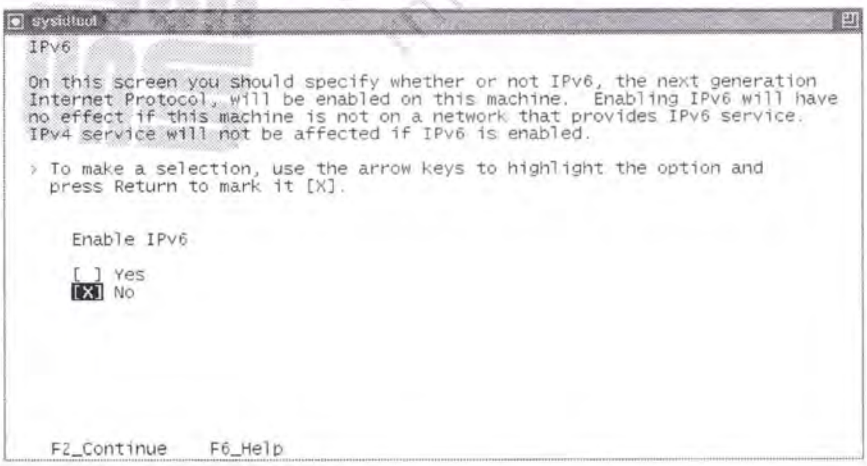

Figure 5-14 IPv6 Window

14. Confirm that the system does not use Internet Protocol version 6. To continue, press F2.

    Figure 5-15 shows the next window that appears. In this window, you can let the operating environment try to find a default route, or you can specify one.

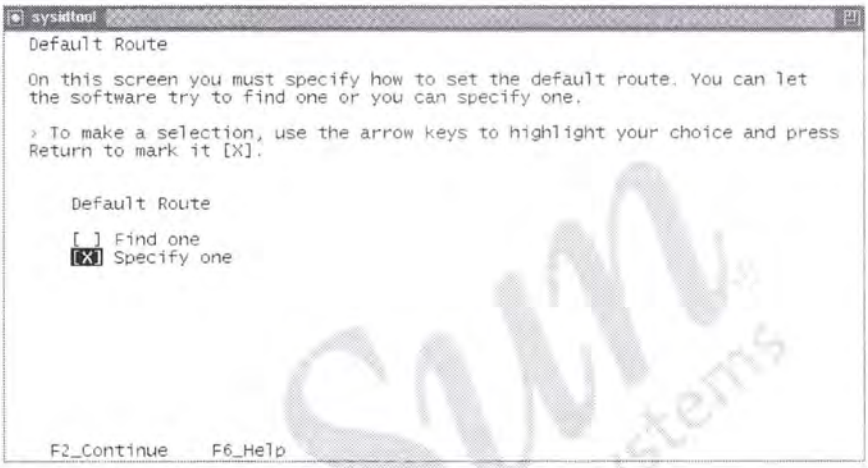

Figure 5-15 Default Route Window

15. Select `Specify one`. To continue, press F2.

Figure 5-16 shows the next window that appears.

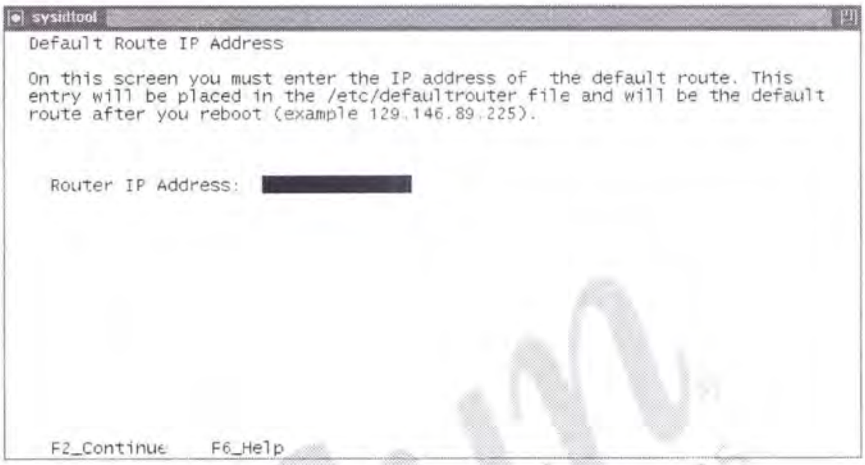

Figure 5-16 Default Route IP Address Window

16. Enter the default route IP address provided to you by your instructor. To continue, press F2.

    Figure 5-17 shows the next window that appears.

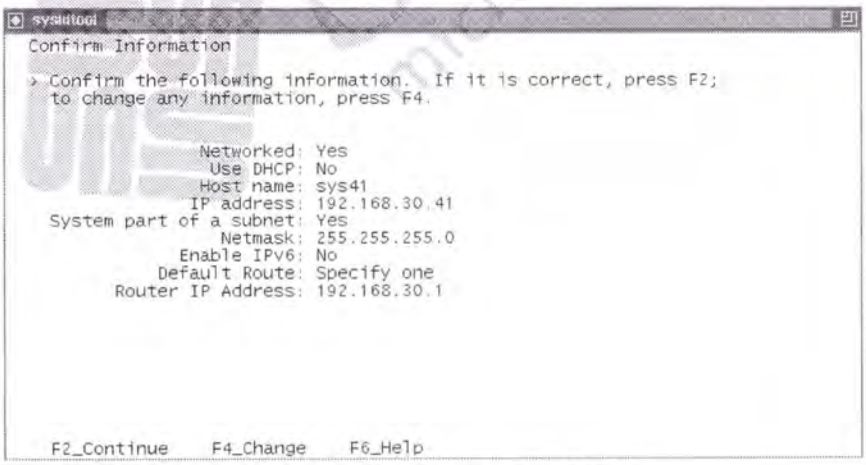

Figure 5-17 Confirm System Identification Window

17. Verify your system configuration. Press F4 to go back and make changes or to correct errors. To continue, press F2.

    Figure 5-18 shows the next window that appears.

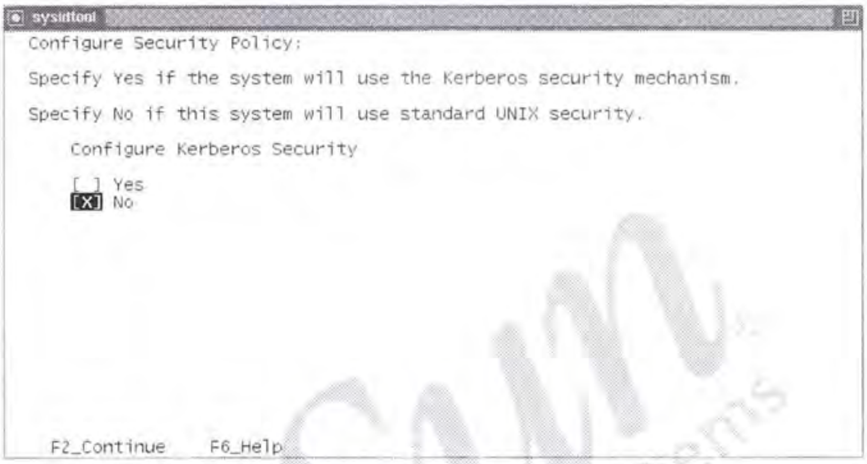

Figure 5-18 Configure Security Policy Window

18. Select No to configure the Solaris 9 OE to use standard UNIX® security. To continue, press F2.

    Figure 5-19 shows the next window that appears.

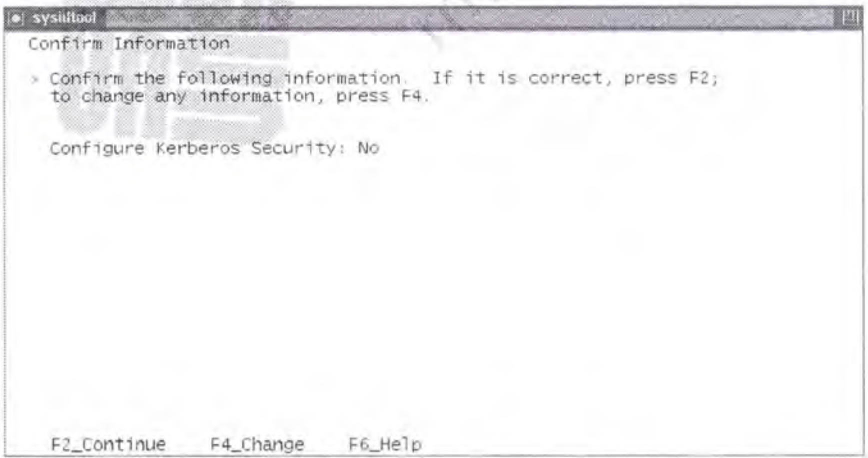

Figure 5-19 Confirm Security Information Window

19. Press F2 to confirm the `No` response and to display the next window.

    Figure 5-20 shows the next window that appears.

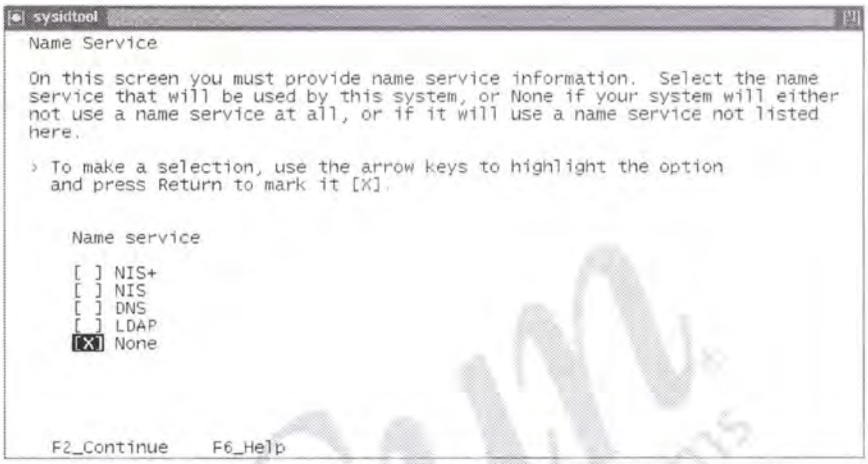

Figure 5-20 Name Service Window

20. Select `None` as your name service. To continue, press F2.

    Figure 5-21 shows the next window that appears.

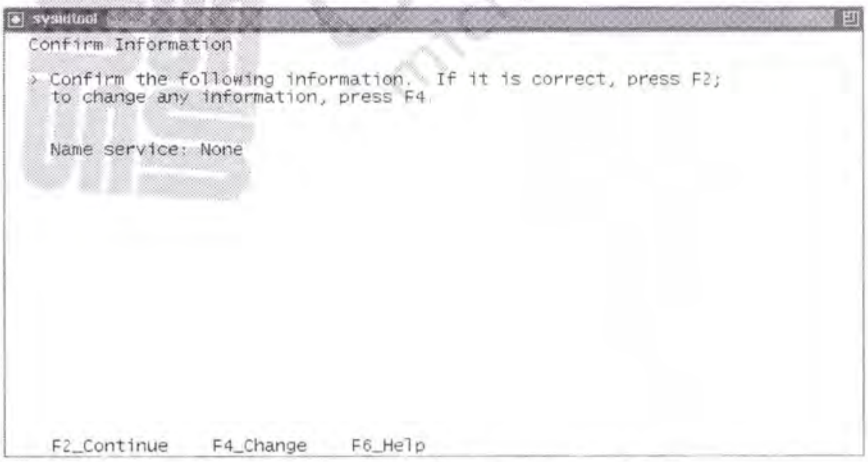

Figure 5-21 Confirm Name Service Information Window

21. To change the information, press F4. To continue, press F2.

    Figure 5-22 shows the next window that appears.

```
sysidtool
Time Zone

On this screen you must specify your default time zone.  You can specify a
time zone in three ways:  select one of the continents or oceans from the
list, select other - offset from GMT, or other - specify time zone file.

> To make a selection, use the arrow keys to highlight the option and
  press Return to mark it [X].

    Continents and Oceans

-   [ ] Africa
    [ ] Americas
    [ ] Antarctica
    [ ] Arctic Ocean
    [ ] Asia
    [ ] Atlantic Ocean
    [ ] Australia
    [ ] Europe
v   [ ] Indian Ocean

   F2_Continue    F6_Help
```

Figure 5-22 Time Zone Window

22. Select the appropriate time zone continent. To continue, press F2.

    Figure 5-23 shows the next window that appears.

```
sysidtool
Country or Region

> To make a selection, use the arrow keys to highlight the option and
  press Return to mark it [X].

    Countries and Regions

-   [ ] United States
    [ ] Anguilla
    [ ] Antigua & Barbuda
    [ ] Argentina
    [ ] Aruba
    [ ] Bahamas
    [ ] Barbados
    [ ] Belize
    [ ] Bolivia
    [ ] Brazil
    [ ] Canada
    [ ] Cayman Islands
v   [ ] Chile

   F2_Continue    F6_Help
```

Figure 5-23 Country or Region Window

23. Select the appropriate time zone country or region. To continue, press F2.

    Figure 5-24 shows the next window that appears.

```
sysidtool
Time Zone

> To make a selection, use the arrow keys to highlight the option and
  press Return to mark it [X].

    Time zones

-   [ ] Eastern Time
    [ ] Eastern Time - Michigan - most locations
    [ ] Eastern Time - Kentucky - Louisville area
    [ ] Eastern Time - Kentucky - Wayne County
    [ ] Eastern Standard Time - Indiana - most locations
    [ ] Eastern Standard Time - Indiana - Crawford County
    [ ] Eastern Standard Time - Indiana - Starke County
    [ ] Eastern Standard Time - Indiana - Switzerland County
    [ ] Central Time
    [ ] Central Time - Michigan - Wisconsin border
    [ ] Central Time - North Dakota - Oliver County
    [X] Mountain Time
v   [ ] Mountain Time - south Idaho & east Oregon

  F2_Continue    F6_Help
```

Figure 5-24 Time Zone Window

24. Select the appropriate time zone for your area. To continue, press F2.

    Figure 5-25 shows the next window that appears.

```
sysidtool
Date and Time

> Accept the default date and time or enter
  new values.

Date and time: 2002-03-20 09:58

  Year   (4 digits) : 2002
  Month  (1-12)     : 03
  Day    (1-31)     : 20
  Hour   (0-23)     : 09
  Minute (0-59)     : 58

  F2_Continue    F6_Help
```

Figure 5-25 Date and Time Window

25. Accept the default date and time, or enter new values. To continue, press F2.

    Figure 5-26 shows the next window that appears.

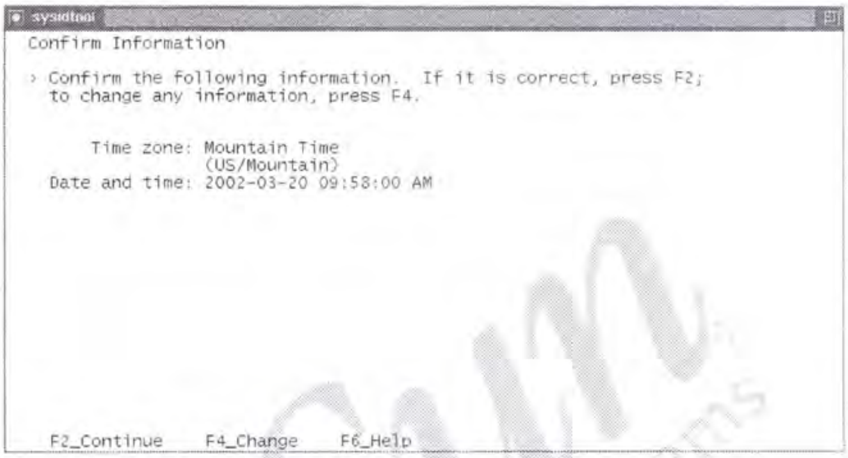

Figure 5-26 Confirm Date and Time Information Window

26. Review the information. To change the information, press F4. To continue, press F2.

    Figure 5-27 shows the next window that appears. The console window displays additional information while generating a software table of contents, checking the `rules.ok` file, and executing scripts.

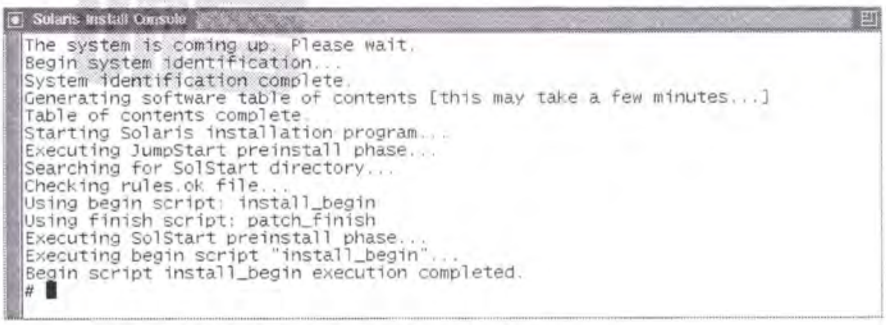

Figure 5-27 Solaris Install Console Window

Figure 5-28 shows the next window that appears.

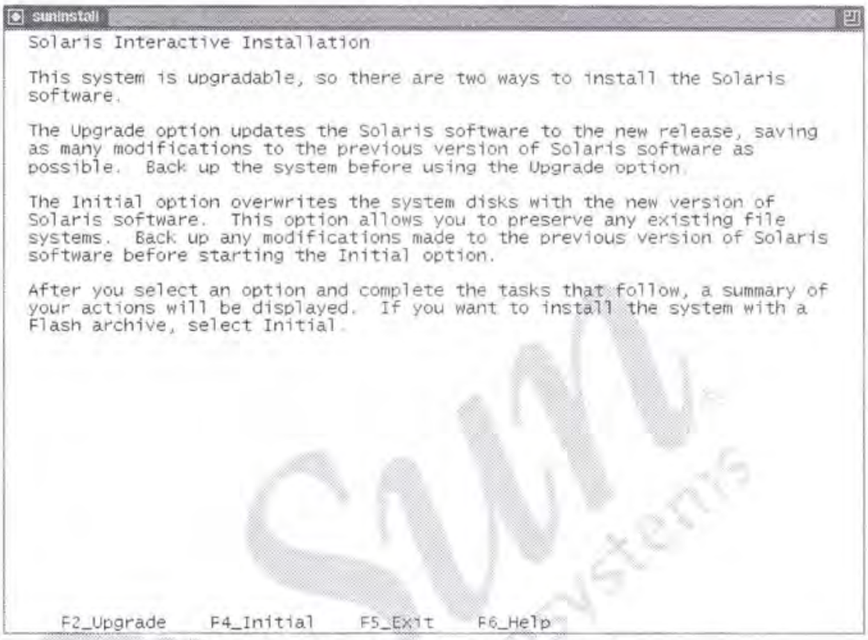

Figure 5-28 Solaris Interactive Installation Window

If you have previously installed a version of the Solaris OE software on the system, the `suninstall` program advises you that the system can be upgraded. The upgrade procedure attempts to preserve local modifications to the system whenever possible. An upgrade procedure generally takes two or three times longer than the initial installation procedure because it does file comparisons.

Press F2 to bypass the initial procedure and to access the upgrade program. The first thing the upgrade program does is analyze the current Solaris OE files and disk configuration. The upgrade program then calculates the size of replacement packages to determine if the disk partitioning is adequate for the new software. If adequate space is allocated, the program prompts you to customize the software for the upgrade.

The upgrade program attempts to mount all file systems listed in the `/etc/vfstab` file. If any file system cannot be mounted, the upgrade program reports the failure and then exits.

If there is no need to preserve existing data on the system, you press F4 to perform the initial installation. The Initial option destroys the existing file systems as it performs an installation of the Solaris 9 OE.

27. For this demonstration, use the Initial installation method. Press F4 to select the Initial installation method.

    Figure 5-29 shows the next window that appears.

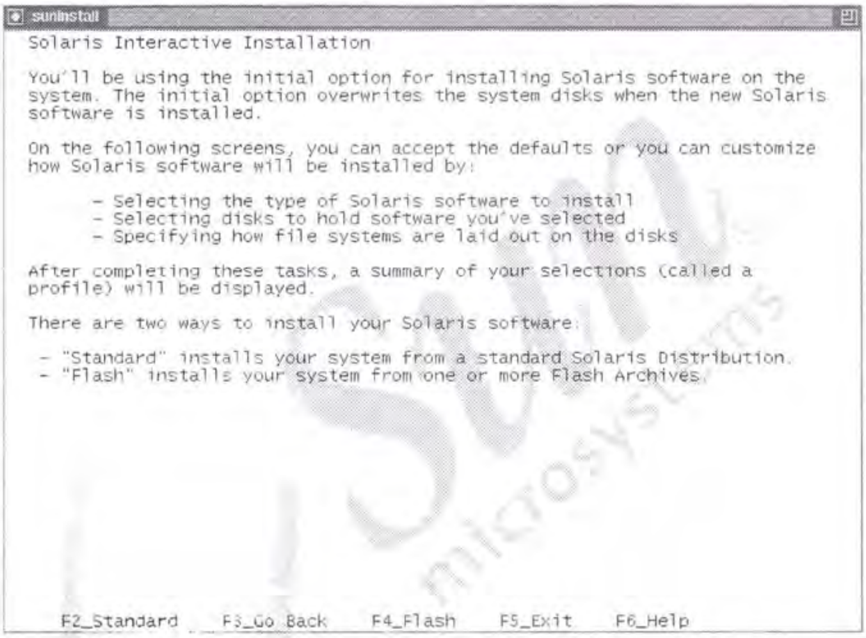

Figure 5-29 Solaris Interactive Installation Window – Initial Option

The Standard method and the Flash method are the two methods available for installing the Solaris 9 OE.

28. Select the Standard method for this demonstration. To continue, press F2.

Figure 5-30 shows the next window that appears.

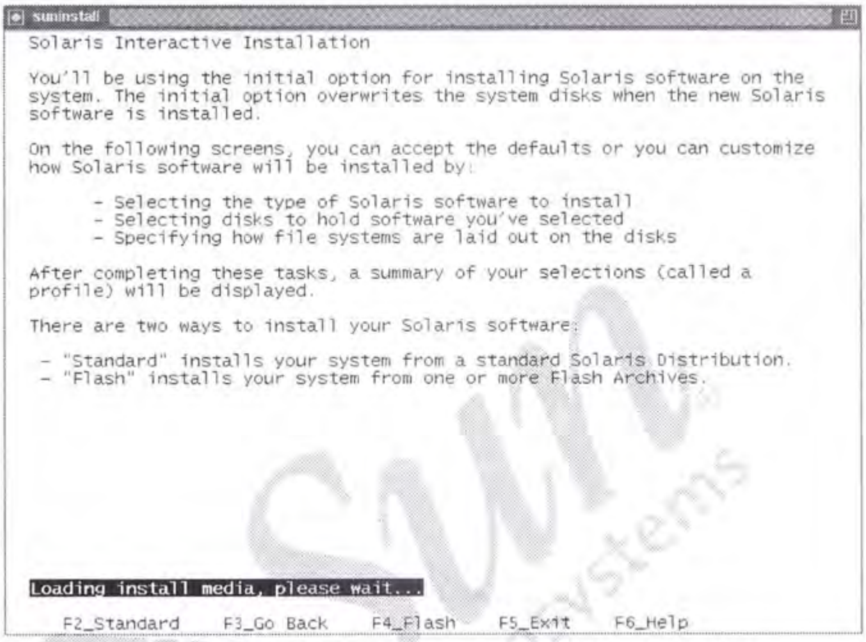

Figure 5-30 Solaris Interactive Installation Window – Loading Installation Media

The Solaris Interactive Installation Loading Install Media window appears briefly to inform you that the `suninstall` program is loading the software.

Figure 5-31 shows the next window that appears.

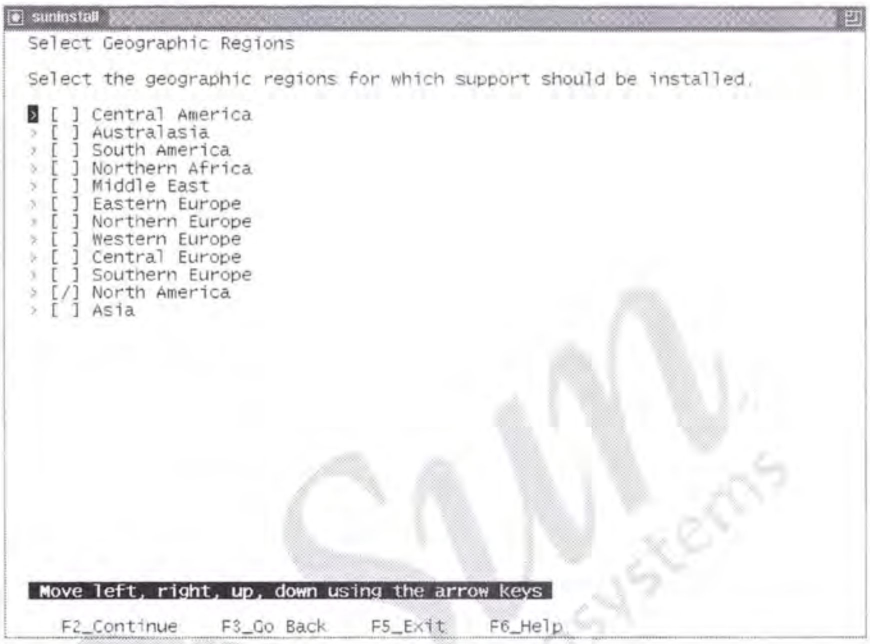

Figure 5-31 Select Geographic Regions Window

Geographic regions are composed of locales and languages. You can select support for a portion of a region or an entire region. You can also select support for more than one region. An X means support for a region or locale is selected. A slash (/) means the region or locale is partially selected.

29. Make the appropriate selections. To continue, press F2.

Figure 5-32 shows the next window that appears.

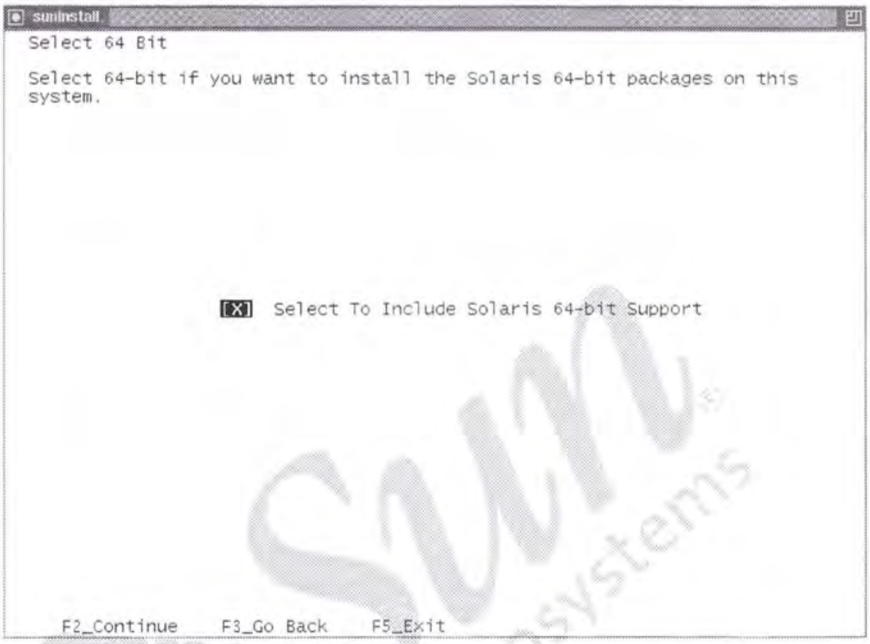

Figure 5-32 Select 64 Bit Window

30. If your system is capable of supporting 64-bit applications, select the available option. To continue, press F2.

Figure 5-33 shows the next window that appears.

Figure 5-33 Select Software

You can select the software group that most closely fits the specific needs of your system. Notice the recommended or estimated disk file size required to install each of the software groups. These sizes vary based on the system type and `kernel` architecture.

31. Select the `Entire Distribution 64-bit`. To continue, press F2.

Figure 5-34 shows the next window that appears.

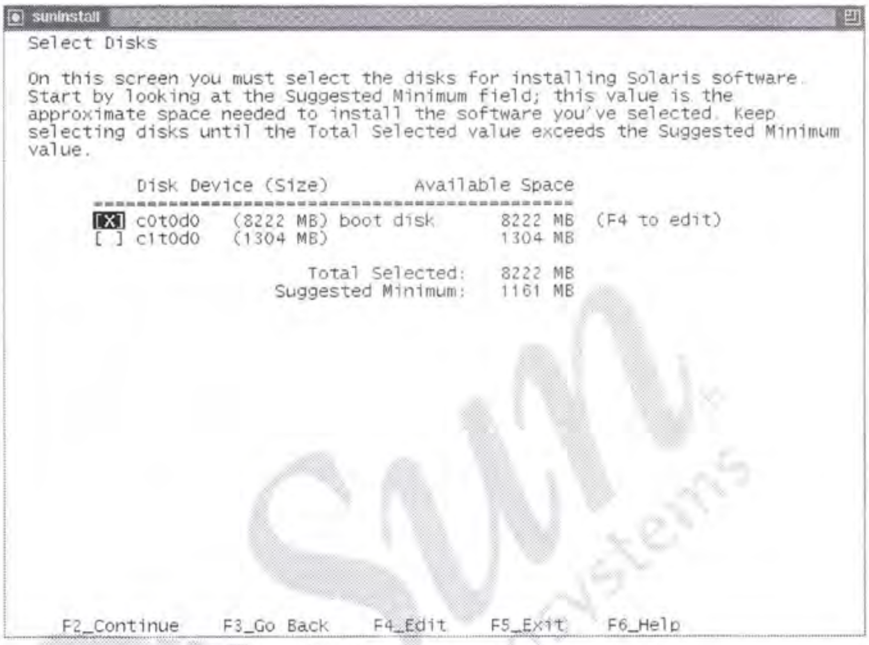

Figure 5-34 Select Disks Window

32. Select the disk or disks on which you are installing the software.

    The window displays values that reflect available space on the disk and the suggested minimum space. Recall that the size of the clusters varies and that there are other general considerations for determining disk slices and sizes. If you choose to change your boot drive, the `suninstall` program prompts you to verify the change and makes changes to your nonvolatile random access memory (NVRAM) parameters.

33. To continue, press F2.

Figure 5-35 shows the next window that appears.

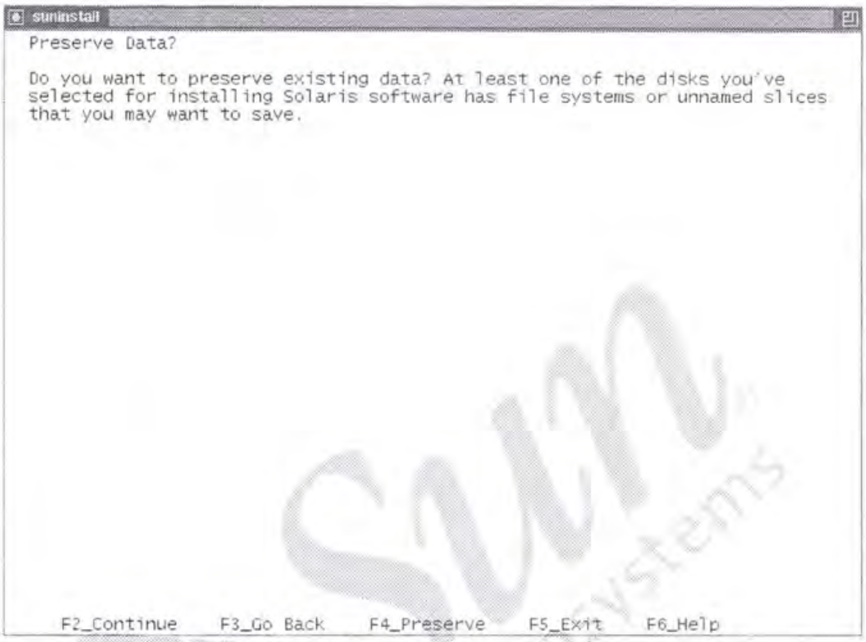

Figure 5-35 Preserve Data Window

The initial installation preserves data only on demand.

34. To continue, press F2.

Note – If you select F4 to preserve data, the `suninstall` program displays a window that enables you to preserve data on a specific partition of the disk. If your system was previously a home directory server, you might want to preserve the `/export/home` file system.

Figure 5-36 shows the next window that appears.

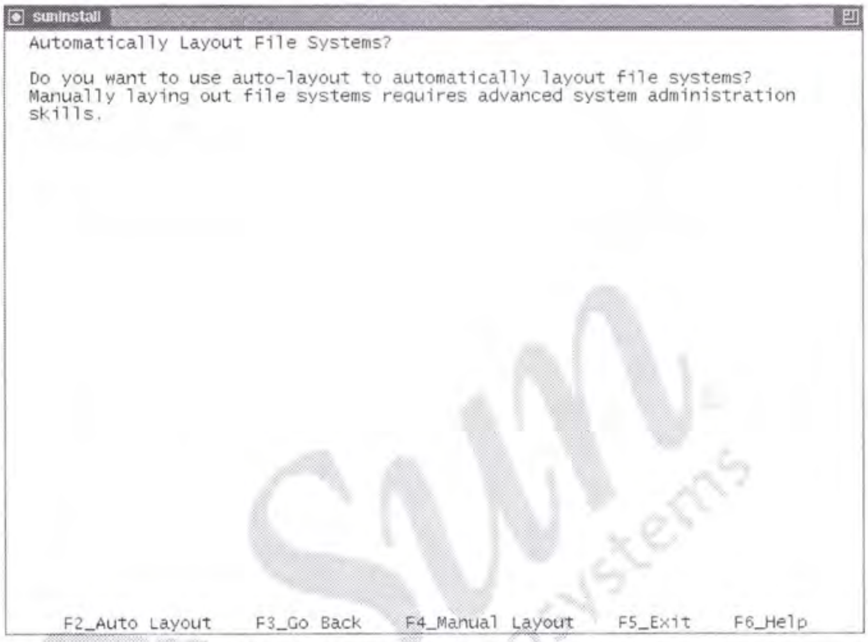

Figure 5-36 Automatic Layout File System Window

The `suninstall` program can automatically lay out the file system arrangement, or you can select disks and slices manually.

35. Press F4 to select `Manual Layout`.

Figure 5-37 shows the next window that appears.

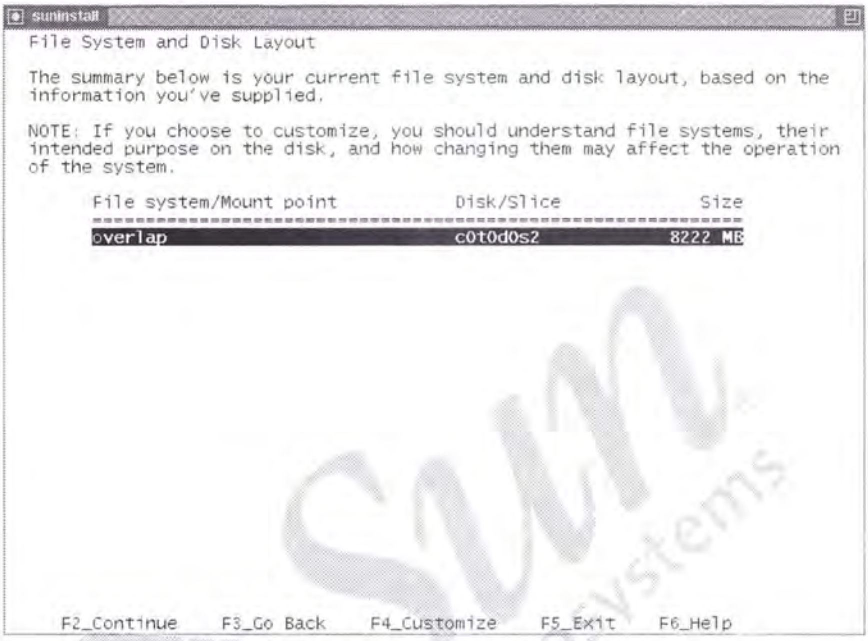

Figure 5-37 File System and Disk Layout Window

This window summarizes the current file system and disk layout. The window generally reflects the overlap partition of the boot drive.

36. Press F4 to select `Customize`.

Figure 5-38 shows the next window that appears.

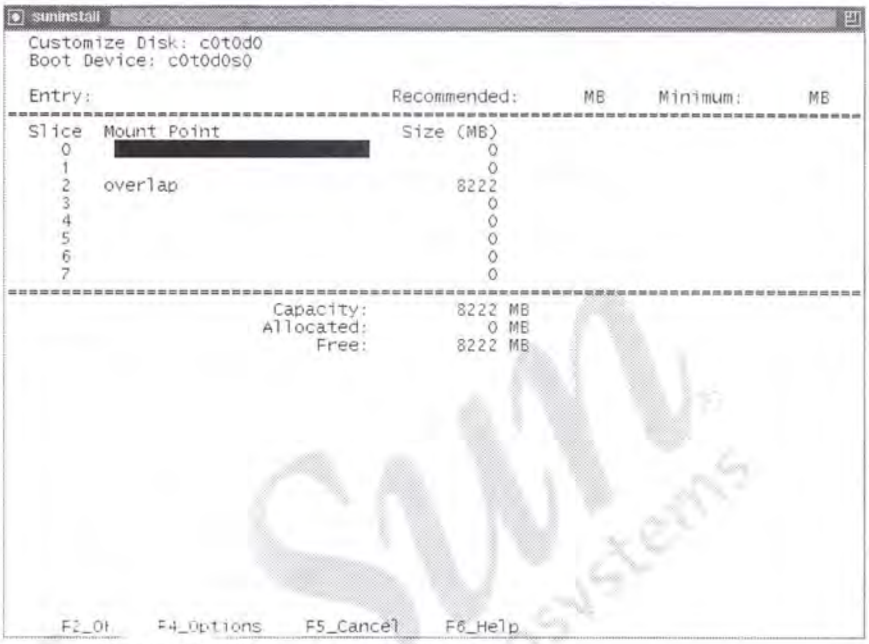

Figure 5-38 Customize Disk Window

The Customize Disk window is a tool you use to reconfigure disk partitions for each disk selected. There are numerous ways to partition slices and to name file systems. Your instructor should inform you of the number of partitions and their sizes for this demonstration.

37. Select the disk slice you want to change. Enter the mount point for the file system that will reside on the slice and the size you want to apply to the slice. Press Return.

    The `Size (MB)` column reflects your changes. The `Allocated` and `Free Space` variables change as you configure each slice of the disk. Recommendations and minimum size requirements are displayed at the upper right.

38. When you have finished reconfiguring the disk, press F4.

Figure 5-39 shows the next window that appears.

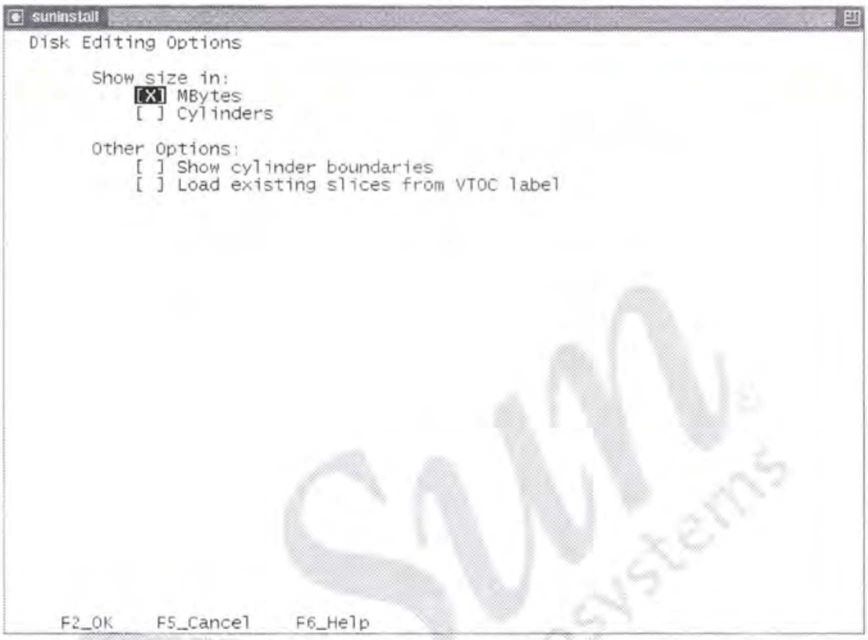

Figure 5-39 Disk Editing Options Window

The Disk Editing Options window enables you to choose how disks are displayed and computed.

39. Make your selections. To continue, press F2.

Figure 5-40 shows the next window that appears.

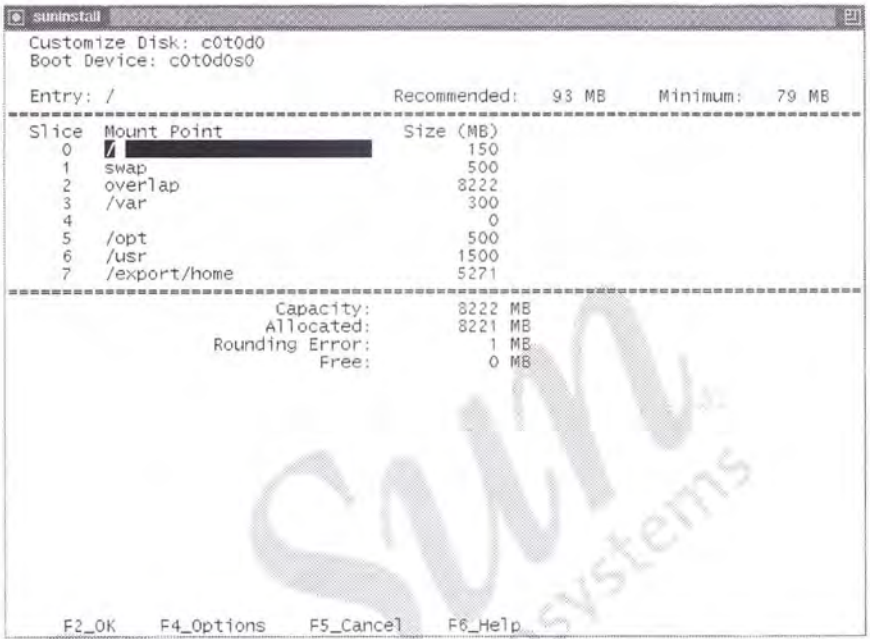

Figure 5-40 Customize Disk Finished Window

The Customize Disk Finished window enables you to review and modify your changes.

40. To continue, press F2.

Figure 5-41 shows the next window that appears

```
suninstall
File System and Disk Layout

The summary below is your current file system and disk layout, based on the
information you've supplied.

NOTE: If you choose to customize, you should understand file systems, their
intended purpose on the disk, and how changing them may affect the operation
of the system.

      File system/Mount point             Disk/Slice             Size
      ================================================================
      /                                   c0t0d0s0              150 MB
      swap                                c0t0d0s1              500 MB
      overlap                             c0t0d0s2             8222 MB
      /var                                c0t0d0s3              300 MB
      /opt                                c0t0d0s5              500 MB
      /usr                                c0t0d0s6             1500 MB
      /export/home                        c0t0d0s7             5271 MB

    F2_Continue    F3_Go Back    F4_Customize    F5_Exit    F6_Help
```

Figure 5-41 File System and Disk Layout Window – Summary

The File System and Disk Layout Summary window is your final confirmation of what the disk layout looks like.

41. To continue, press F2.

Figure 5-42 shows the next window that appears.

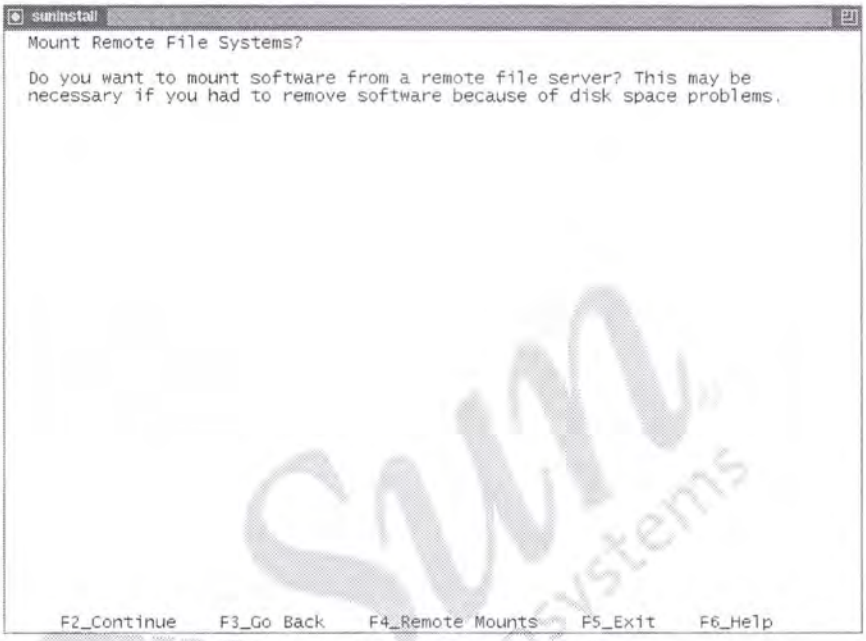

Figure 5-42 Mount Remote File Systems Window

42. Press F4 to open a window that prompts you to enter a server name, an IP address, and a mount point to a location where you have stored data. To continue, press F2.

Figure 5-43 shows the next window that appears.

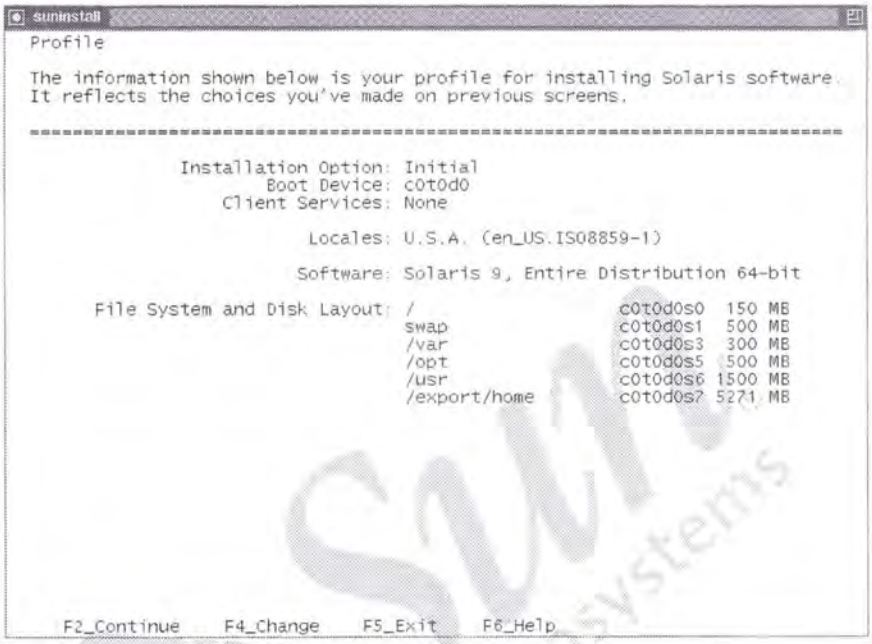

Figure 5-43 Profile Window

The Profile window displays the installation choices you made in previous windows.

This is the last window that enables you to change the options you have selected.

43. When you are satisfied with your selections, press F2.

Figure 5-44 shows the next window that appears.

Figure 5-44 Reboot After Installation Window

The Reboot After Installation window enables you to choose between an auto or a manual reboot.

44. After making your selection, press F2 to begin the installation process.

    The system begins the installation by writing a Volume Table of Contents (VTOC) on the disk or disks selected and creating file systems.

```
Preparing system for Solaris install

Configuring disk (c0t0d0)
        - Creating Solaris disk label (VTOC)

Creating and checking ufs file systems
        - Creating / (c0t0d0s0)

Beginning Solaris Installation
```

Figure 5-45 shows the next window that appears.

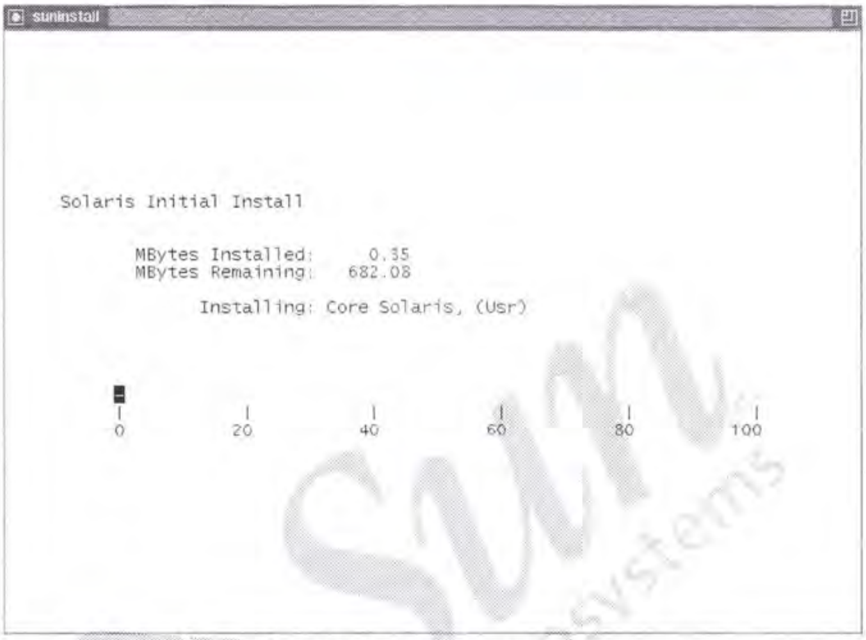

Figure 5-45 Solaris Initial Install Window

The Solaris Initial Install window displays the software cluster currently being installed. The window indicates how many megabytes of the cluster have been installed and how many megabytes of the cluster remain to be installed.

At the end of the installation of the Solaris 9 CD-ROM 1 of 2, the system reboots and displays guidelines for creating a `root` password:

```
On this screen you can create a root password.

A root password can contain any number of characters, but only the first
eight characters in the password are significant. (For example, if you
enter 'a1b2c3d4e5f6' as your root password, then 'a1b2c3d4' could also
be used to gain root access.)

You will be prompted to type the root password twice; for security, the
password will not be displayed on the screen as you type it.
```

```
> If you do not want a root password, press RETURN twice.

Root password:

Press Return to continue.
```

45. Use cangetin as the root password on the newly installed system.

    If your hardware supports it, the system displays a message that asks you if you want to select the automatic power-saving feature.

```
System identification is completed.

     ******************************************************
     This system is configured to conserve energy.
     ******************************************************
     After 30 minutes of idle time on this system, your system state
     will automatically be saved to disk, and the system will power-off.

     Later, when you want to use the system again, and you turn the power
     back on, your system will be restored to its previous state,
     including all the programs that you were running.

     Do you want this automatic power-saving shutdown?
     (If this system is used as a server, answer n) [y,n,?] n
```

46. Type n to disable automatic power saving.

    The system responds with the following message:

```
Autoshutdown has been disabled.

Do you want the system to ask you about this again, when you next reboot?
(This gives you the chance to try it before deciding whether to keep it.)
[y,n,?] n
```

47. Type n to confirm permanent disabling of the automatic power-saving shutdown feature.

    The system responds with the following message:

```
The system will not ask you again about automatic shutdown.

The "Using Power Management" AnswerBook describes more about how
to change and set workstation energy-saving features.
```

During the next phase of the installation, the CDE starts, and the remainder of the windows are displayed.

Figure 5-46 shows the next window that appears.

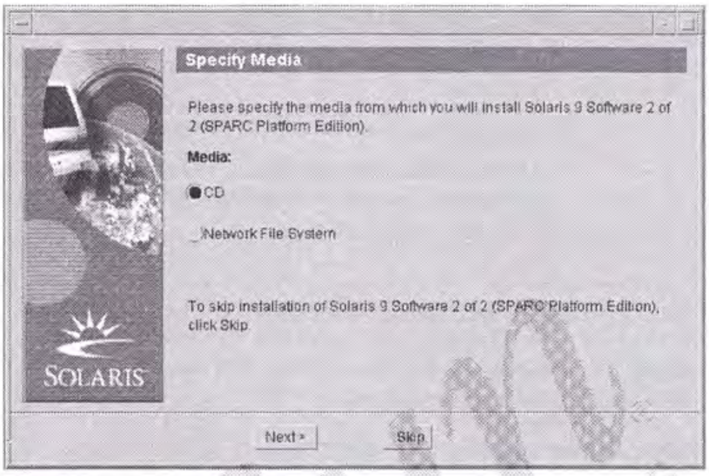

Figure 5-46 Specify Media Window

48. Select the CD option. To continue, click Next.

    Figure 5-47 shows the next window that appears.

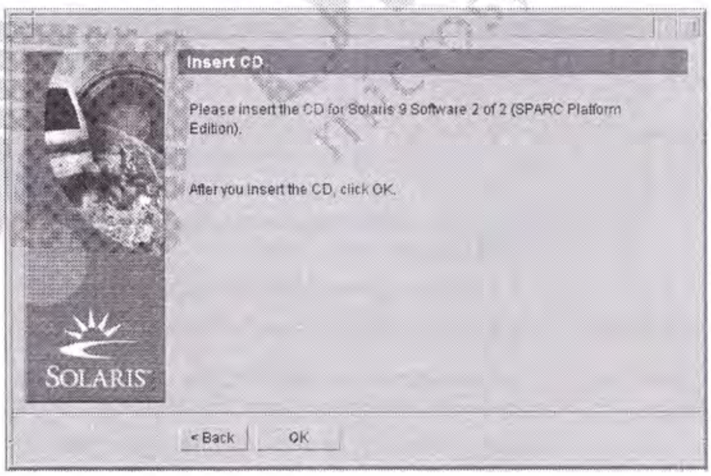

Figure 5-47 Insert CD Window

49. Insert the Solaris 9 OE CD-ROM 2 of 2. To continue, click OK.

Figure 5-48 shows the next window that appears.

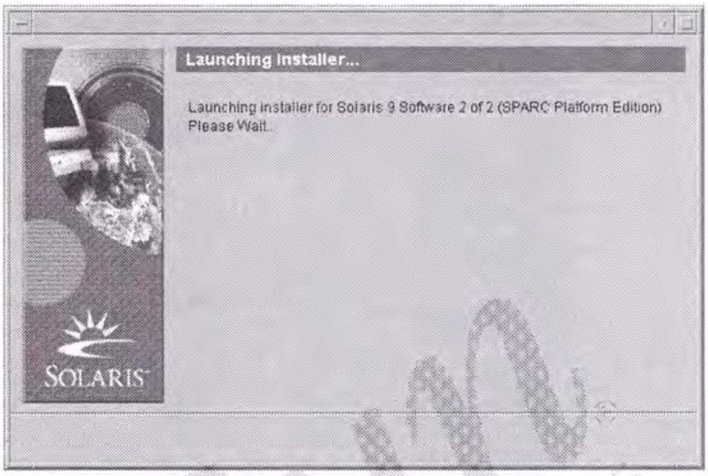

Figure 5-48 Launching Installer Window

The Launching Installer window displays for a moment.

Figure 5-49 shows the next window that appears.

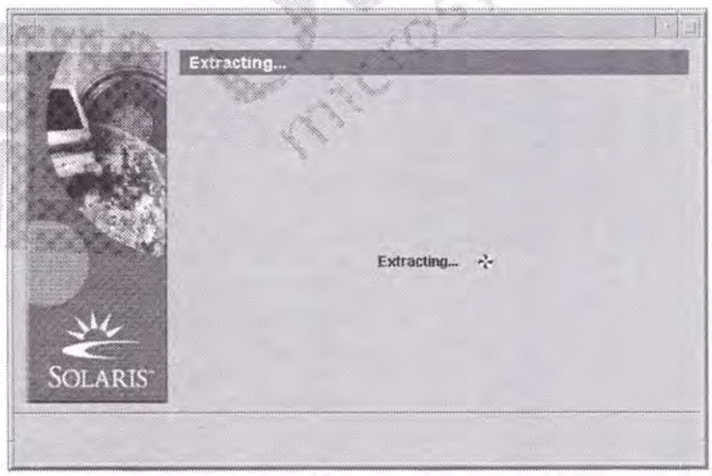

Figure 5-49 Extracting Window

The Extracting window displays a message while the CD-ROM is read.

Figure 5-50 shows the next window that appears.

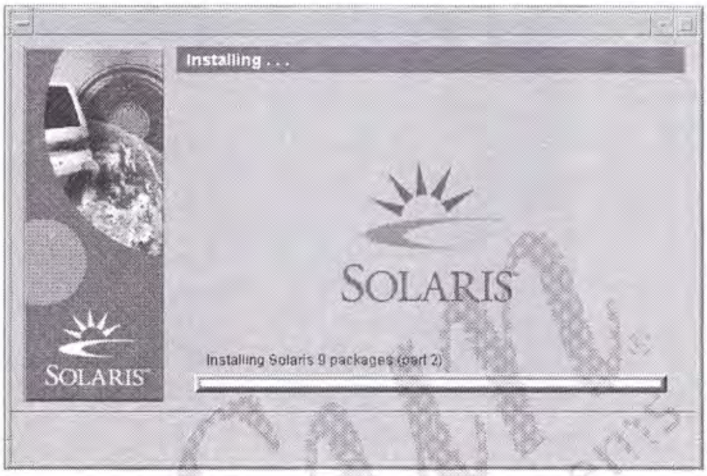

Figure 5-50 Installing Window

While the system installs the packages from the second CD-ROM, the slide bar displays the progress of the installation.

Figure 5-51 shows the next window that appears.

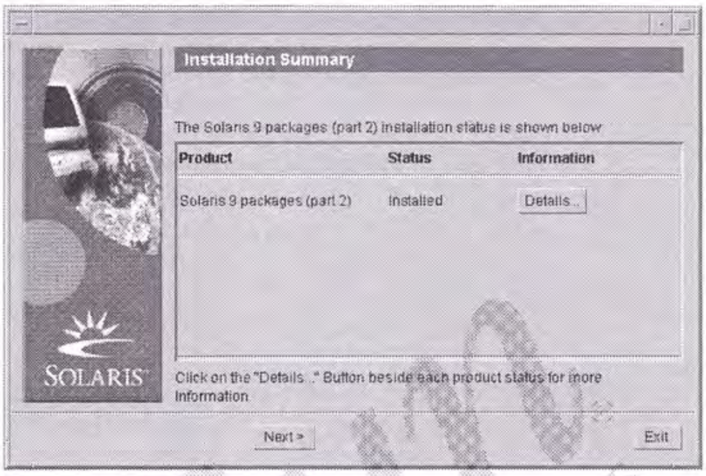

Figure 5-51 Installation Summary Window

50. Click Details to display the log file of the installation. The log file contains information about the packages installed.
51. Click Next when you are finished reviewing the log file.

    Figure 5-52 shows the next window that appears.

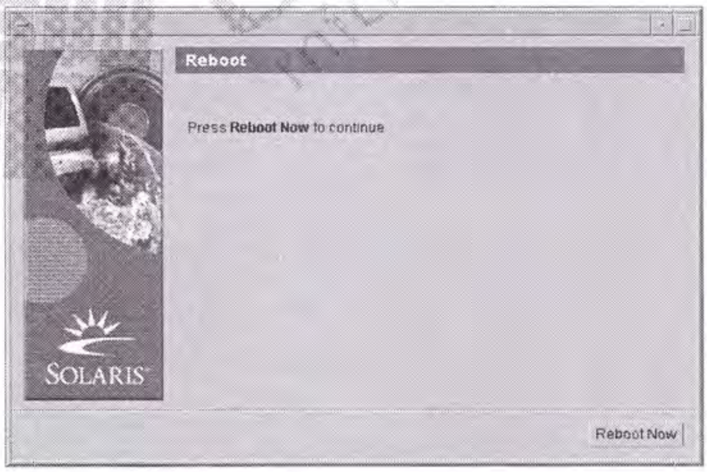

Figure 5-52 Reboot Window

52. Click Reboot Now to reboot the system.
53. After the system completes the reboot process, log in and verify that the system is operational. You can review additional log file information after the system has rebooted by looking at the `/var/sadm/install_data/install_log` file.

You have now completed the installation demonstration.

Module 6

Performing Solaris 9 OE Package Administration

Objectives

Upon completion of this module, you should be able to:

- Describe the fundamentals of package administration
- Administer packages using the command-line interface

The following course map shows how this module fits into the current instructional goal.

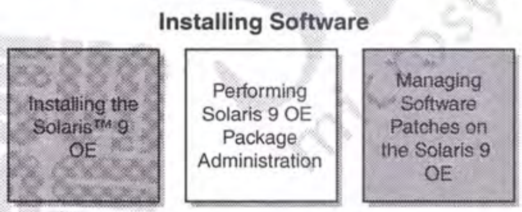

Figure 6-1 Course Map

Introducing the Fundamentals of Package Administration

Software package administration adds software to systems and removes software from systems. Sun and its third-party vendors deliver software products to users in software packages.

Software Packages

The term package refers to the method of distributing software products and installing them in systems. In its simplest form, a package is a collection of files and directories.

Note – The Solaris OE software installation process installs all the required software packages automatically, based on the software group configuration choice.

Software packages contain:

- Files that describe the package and the amount of disk space required for installation
- Compressed software files to be installed on the system
- Optional scripts that run when the package is added or removed

The `/var/sadm/install/contents` File

The `/var/sadm/install/contents` file is a complete record of all the software packages installed on the local system disk. It references every file and directory belonging to every software package and shows the configuration of each product installed. To list the contents of the `/var/sadm/install/contents` file, perform the command:

```
# more /var/sadm/install/contents
(output edited for brevity)
/bin=./usr/bin s none SUNWcsr
/dev d none 0755 root sys SUNWcsr SUNWcsd
/dev/allkmem=../devices/pseudo/mm@0:allkmem s none SUNWcsd
/dev/arp=../devices/pseudo/arp@0:arp s none SUNWcsd
/etc/ftpd/ftpusers e ftpusers 0644 root sys 185 15198 1008137961 SUNWftpr
/etc/passwd e passwd 0644 root sys 467 38912 1008137367 SUNWcsr
```

The pkgadd command updates the /var/sadm/install/contents file each time new packages are installed.

The pkgrm command uses the /var/sadm/install/contents file to determine where the files for a software package are located on the system. When a package is removed from the system, the pkgrm command updates the /var/sadm/install/contents file.

To determine if a particular file was installed on the system disk and to find the directory in which it is located, use the grep command to query the /var/sadm/install/contents file. For example, to verify that the showrev command is installed on the system disk, perform the command:

```
# grep showrev /var/sadm/install/contents
/usr/bin/showrev f none 0755 root sys 29040 1417 993448469 SUNWadmc
/usr/share/man/sman1m/showrev.1m f none 0444 root bin 6416 64169
993108127 SUNWman
```

Administering Packages From the Command Line

From the command line, you can add, remove, check the state of, and display information about packages.

The command-line tools for viewing software, adding software, and removing software from a workstation after the Solaris OE software is installed on a system include:

pkgadd	Installs software packages to the system
pkgrm	Removes a package from the system
pkginfo	Displays software package information
pkgchk	Checks package installation state

Note – Students need to insert the Solaris 9 Software 1 of 2 CD-ROM to demonstrate the steps described in this module.

Displaying Information About Installed Software Packages

Use the pkginfo command to display information about the software packages installed on the local system's disk. The /var/sadm/pkg directory maintains a record of all installed packages.

For example, to display information about software packages installed on the local system's disk, perform the command:

```
# pkginfo | more
<some output omitted>
application NSCPcom      Netscape Communicator
system      SUNWaccr     System Accounting, (Root)
system      SUNWaccu     System Accounting, (Usr)
system      SUNWadmap    System administration applications
system      SUNWadmc     System administration core libraries
system      SUNWaudd     Audio Drivers)
ALE         SUNWciu8     Simplified Chinese iconv modules for UTF-8
system      SUNWcsd      Core Solaris Devices
CTL         SUNWctplx    Portable layout services for CTL (64-bit)
system      SUNWdoc      Documentation Tools
application SUNWdej2p    Java Plug-in
```

The column on the left displays the package category, such as application, system, Complex Text Layout (CTL), or Alternate Language Environment (ALE). A CTL language is any language which stores text differently than it is displayed. An ALE is an alternate language, different from the basic Solaris OE languages.

The center column displays the software package name. If it begins with SUNW, it is a Sun Microsystems product. Otherwise, it represents a third-party package.

The column on the right displays a brief description of the software product.

Displaying Information for All Packages

To display all the available information about the software packages, use the pkginfo command with the -l option.

For example, to view additional information about each software package installed on the local systems hard drive, perform the command:

```
# pkginfo -l | more
(output omitted)
```

Displaying Information for a Specific Package

To display the information for a specific software package, specify its name on the command line.

For example, to view the information for the SUNWman software package, perform the command:

```
# pkginfo -l SUNWman

   PKGINST:  SUNWman
      NAME:  On-Line Manual Pages
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  42.0,REV=35
   BASEDIR:  /usr
    VENDOR:  Sun Microsystems, Inc.
      DESC:  System Reference Manual Pages
    PSTAMP:  tinkertoym21003318
  INSTDATE:  Sep 27 2001 10:43
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:   7033 installed pathnames
                 3 shared pathnames
                84 directories
             81450 blocks used (approx)
```

The last line identifies the size of the package. The number of blocks used defines how much space is needed on the disk to install the package.

Note – A block is a 512-byte disk block.

To determine how many packages are currently installed on disk, perform the command:

```
# pkginfo | wc -l
     657
```

Displaying Information for Software Packages

To view information about packages that are located on Solaris 9 Software 1 of 2 CD-ROM, perform the command:

```
pkginfo -d /cdrom/cdrom0/s0/Solaris_9/Product |more
```

Software groups located on Solaris 9 Software 1 of 2 CD-ROM are Core and End User.

To view information about packages that are located on Solaris 9 Software 2 of 2 CD-ROM, perform the command:

```
pkginfo -d /cdrom/cdrom0/Solaris_9/Product |more
```

The software groups located on Solaris 9 Software 2 of 2 CD-ROM are the Developer, Entire Distribution, and Entire Distribution Plus OEM Support software groups.

Adding a Software Package

When you add a software package, the pkgadd command copies the files from the installation media to the local system's disk and executes scripts to uncompress files. By default, the pkgadd command requires confirmation during the package add process.

For example, to transfer the SUNWns6m software package from a CD-ROM and install it on the system, perform the commands:

```
# cd /cdrom/cdrom0/Solaris_9/ExtraValue/EarlyAccess/Netscape_6/Packages
# pkgadd -d . SUNWns6m
Processing package instance <SUNWns6m> from
<cdrom/sol_9_sparc_2/Solaris_9/ExtraValue/EarlyAccess/Netscape_6/Packages
>

Netscape 6 for Solaris - Messenger
(sparc) 6.2,REV=20.2002.03.06
Copyright 2002 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
Using </usr> as the package base directory.
## Processing package information.
## Processing system information.
   2 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.
```

```
Do you want to continue with the installation of <SUNWns6m> [y,n,?] y

Installing Netscape 6 for Solaris - Messenger as <SUNWns6m>

## Installing part 1 of 1.
15038 blocks

Installation of <SUNWns6m> was successful.
```

Note – Certain unbundled and third-party packages are no longer entirely compatible with the latest version of the pkgadd command. These packages require system administrator interaction throughout the installation and not just at the very beginning. To install these older packages set the following environment variable: NONABI_SCRIPTS=TRUE.

Checking a Package Installation

The pkgchk command checks to determine if a package has been completely installed on the system. The pkgchk command also checks the path name, the file size and checksum, and the file attributes of a package. If the pkgchk command does not display a message, it indicates the package was installed successfully and that no changes have been made to any files or directories in the package.

The following example checks the contents and attributes of the SUNWcarx software package currently installed on the system.

```
# pkgchk SUNWcarx
#
```

To list the files contained in a software package, use the -v option.

For example, to list the files in the SUNWcarx software package, perform the command:

```
# pkgchk -v SUNWcarx
(some output omitted)
/kernel
/kernel/drv
/kernel/drv/sparcv9
/kernel/drv/sparcv9/arp
/var
```

```
/var/ld
/var/ld/64
/var/ld/sparcv9
```

To determine if the contents and attributes of a file have changed since it was installed with its software package, use the -p option.

For example, to check the /etc/shadow file, perform the command:

```
# pkgchk -p /etc/shadow
ERROR: /etc/shadow
    modtime <06/19/01 03:55:23 PM> expected <11/07/01 02:31:51 PM> actual
    file size <233> expected <247> actual
    file cksum <15950> expected <17155> actual
```

The differences in modtime, file size, and checksum indicate that the original /etc/shadow file has changed in size since the initial Solaris OE software installation.

The -l option with the pkgchk command lists information about selected files that make up a package.

For example, to list information about the /usr/bin/showrev file, perform the command:

```
# pkgchk -l -p /usr/bin/showrev
Pathname: /usr/bin/showrev
Type: regular file
Expected mode: 0755
Expected owner: root
Expected group: sys
Expected file size (bytes): 29040
Expected sum(1) of contents: 1417
Expected last modification: Jun 24 11:54:29 PM 2001
Referenced by the following packages:
        SUNWadmc
Current status: installed
```

The full path must be typed for the pkgchk command to return information about the file.

For example, the pkgchk command does not return any information if the /usr/bin/ path is removed from the previous example.

```
# pkgchk -l -p showrev
#
```

Removing a Software Package

The pkgrm command removes a software package from the system and deletes all of the files associated with that package, unless other packages share those files.

By default, the pkgrm command requires confirmation to continue removing a package and issues a message to warn about possible package dependencies. If package dependencies do exist, the command again requires confirmation to continue with the package removal process.

The following command removes the SUNWapchr software package from the system.

```
# pkgrm SUNWapchr

The following package is currently installed:
   SUNWapchr        Apache Web Server (root)
                    (sparc) 11.9.0,REV=2002.01.18.00.45

Do you want to remove this package? [y,n,?,q] y

## Removing installed package instance <SUNWapchr>
## Verifying package dependencies.
WARNING:
    The <SUNWapchu> package depends on the package
    currently being removed.
WARNING:
    The <SUNWapchd> package depends on the package
    currently being removed.
Dependency checking failed.

Do you want to continue with the removal of this package [y,n,?,q] y
## Processing package information.
## Removing pathnames in class <initd>
/etc/rcS.d/K16apache
/etc/rc3.d/S50apache
/etc/rc2.d/K16apache
/etc/rc1.d/K16apache
```

```
/etc/rc0.d/K16apache
/etc/init.d/apache

(output ommited for brevity)

## Removing pathnames in class <preserve>
/var <shared pathname not removed>
/etc/rcS.d <shared pathname not removed>
/etc/rc3.d <shared pathname not removed>
/etc/rc2.d <shared pathname not removed>
/etc/rc1.d <shared pathname not removed>
/etc/rc0.d <shared pathname not removed>
/etc/init.d <shared pathname not removed>
/etc/apache/tomcat/server.xml-example
/etc/apache/tomcat
/etc/apache/httpd.conf-example
/etc/apache/README.Solaris
/etc/apache
/etc <shared pathname not removed>
## Updating system information.

Removal of <SUNWapchr> was successful.
```

Note – A file shared by two or more packages displays the message `filename <shared pathname not removed>`. The message is removed only when the file is no longer shared.

Adding Packages by Using a Spool Directory

For convenience, copy frequently installed software packages from the Solaris 9 Software CD-ROM to a spool directory on the system.

The default installation directory for packages that have been spooled, but not installed, is `/var/spool/pkg`. The `pkgadd` command, by default, looks in the `/var/spool/pkg` directory for any packages specified on the command line.

Copying packages from the CD-ROM into a spool directory is not the same as installing the packages on disk.

To copy a package into the /var/spool/pkg directory, perform the command:

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product -s spool SUNWensqr.u
Transferring <SUNWensqr.u> package instance
```

The -s option with the keyword spool copies the package into the /var/spool/pkg directory by default.

To verify that the package exists in the spool directory, perform the command:

```
# ls -al /var/spool/pkg
total 6
drwxrwxrwt   3 root     bin          512 Nov  7 13:53 .
drwxr-xr-x  10 root     bin          512 Oct 25 09:01 ..
drwxr-xr-x   5 root     other        512 Nov  7 13:53 SUNWensqr.u
```

To remove software packages from a spool directory, use the pkgrm command with the -s option.

```
# pkgrm -s spool SUNWensqr.u
The following package is currently spooled:
SUNWensqr.u  Ensoniq ES1370/1371/1373 Audio Device Driver (Root),(32-bit)
             (sparc.sun4u) 6.1,REV=2001.05.04.09.44

Do you want to remove this package? y

Removing spooled package instance <SUNWensqr.u>
```

If alternative spooling directories exist, specify which directory to use by adding a directory path to the -s option.

For example, to select the /export/pkg directory, perform the commands:

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product -s /export/pkg SUNWensqr.u

# pkgrm -s /export/pkg SUNWensqr.u
```

Reviewing Package Administration

This section details the package administration tasks.

Table 6-1 summarizes the commands used for package administration.

Table 6-1 Package Administration Commands

Command Name	Description
pkginfo	Lists packages installed on the system or available on distribution media
pkgadd	Installs packages
pkgrm	Removes packages
pkgchk	Verifies the attributes of the path names that belong to packages

Table 6-2 summarizes the files and directories used in package administration.

Table 6-2 Package Administration Files and Directories

File or Directory	Description
/var/sadm/install/contents	A software package map of the entire system
/opt/*pkgname*	The preferred location for the installation of unbundled packages
/opt/*pkgname*/bin or /opt/bin	The preferred location for the executable files of unbundled packages
/var/opt/*pkgname* or /etc/opt/*pkgname*	The preferred location for log files of unbundled packages

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Manipulating Software Packages (Level 1)

In this exercise, you use package-related commands to remove, install, and spool packages.

Preparation

Locate the Solaris 9 Software CD-ROMs. Refer to the lecture notes as necessary to perform the tasks listed.

Tasks

Complete the following tasks:

- Find the names of packages installed on your system that relate to manuals. List and record the status of, the install date of, the number of files used by, and the number of blocks used by the `SUNWman` package. Obtain the same information from the spooled `SUNWman` package on the correct Solaris 9 OE Software CD-ROM. Remove and reinstall the `SUNWman` package.

  (Steps 1–6 in the Level 2 lab)

- Remove the `SUNWdoc` package from the system. Attempt to access the online man pages. Spool the `SUNWdoc` package from the correct Solaris 9 OE software CD-ROM into the default spool area. Verify the presence of this package in the spool area. Add the `SUNWdoc` package to the system. Remove the `SUNWdoc` package from the spool area.

  (Steps 7–15 in the Level 2 lab)

Exercise: Manipulating Software Packages (Level 2)

In this exercise, you use package-related commands to remove, install, and spool packages.

Preparation

Locate the Solaris 9 Software CD-ROMs. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Find the names of packages installed on your system that relate to manuals. List and record the status of, the install date of, the number of files used by, and the number of blocks used by the `SUNWman` package. Obtain the same information from the spooled `SUNWman` package on the correct Solaris 9 OE Software CD-ROM. Remove and reinstall the `SUNWman` package.
- Remove the `SUNWdoc` package from the system. Attempt to access the online man pages. Spool the `SUNWdoc` package from the correct Solaris 9 OE software CD-ROM into the default spool area. Verify the presence of this package in the spool area. Add the `SUNWdoc` package to the system. Remove the `SUNWdoc` package from the spool area.

Tasks

Complete the following steps:

1. Insert the Solaris 9 Software 2 of 2 CD-ROM SPARC Platform Edition into the drive.
2. Use the `pkginfo` command to search for packages currently on your system that are related to manuals.

   Which packages were listed?
3. Display a long-format listing of the information for the `SUNWman` package installed on your system. What is listed for the status of, the install date of, the number of files used by, and the number of blocks used by this package?

4. Display a long-format listing of the information for the `SUNWman` package on the Solaris 9 OE Software 2 of 2 CD-ROM. Obtain the same information as in the previous step.

Note – Steps 5 and 6 take several minutes to perform.

---

5. Remove the `SUNWman` package from your system, and verify that it has been removed by trying to access the manual pages.
6. Reinstall the `SUNWman` package from the Solaris 9 OE Software 2 of 2 CD-ROM. Respond `y` to questions asked by the `pkgadd` command. Verify that the manual pages work.
7. Remove the `SUNWdoc` package from your system and answer yes to the remove questions.
8. Are there any package dependencies related to removing this package?
9. Eject the Solaris 9 Software 2 of 2 CD-ROM, and insert the Solaris 9 Software 1 of 2 CD-ROM. Use the `pkgadd` command to spool the `SUNWdoc` package into the default spool area.
10. Use the `pkginfo` command with the appropriate options to verify the presence of the `SUNWdoc` package in the default spool area.
11. Install the `SUNWdoc` package. Observe the messages, and verify that the package is installed from the `/var/spool/pkg` directory.
12. Remove the `SUNWdoc` package from the default spool area.
13. Verify that the `SUNWdoc` package no longer exists in the spool area and that it is installed on your system.
14. Eject the Solaris 9 Software 1 of 2 CD-ROM.

Exercise: Manipulating Software Packages (Level 3)

In this exercise, you use package-related commands to remove, install, and spool packages.

Preparation

Locate the Solaris 9 Software CD-ROMs. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Find the names of packages installed on your system that relate to manuals. List and record the status of, the install date of, the number of files used by, and the number of blocks used by the `SUNWman` package. Obtain the same information from the spooled `SUNWman` package on the correct Solaris 9 OE Software CD-ROM. Remove and reinstall the `SUNWman` package.
- Remove the `SUNWdoc` package from the system. Attempt to access the online man pages. Spool the `SUNWdoc` package from the correct Solaris 9 OE software CD-ROM into the default spool area. Verify the presence of this package in the spool area. Add the `SUNWdoc` package to the system. Remove the `SUNWdoc` package from the spool area.

Tasks and Solutions

Complete the following steps:

1. Insert the Solaris 9 Software 2 of 2 CD-ROM SPARC Platform Edition into the drive.
2. Use the `pkginfo` command to search for packages currently on your system that are related to manuals.

```
# pkginfo | grep anual
```

Which packages were listed?

`SUNWman`, `SUNWmfman`, `SUNWopl5m`, `SUNWpl5m`, *and* `SUNWtltkm`

These packages contain the online manual pages, CDE motif manuals, Perl Reference manual pages, Perl 5 online manual pages, and ToolTalk™ software manual pages, respectively.

3. Display a long-format listing of the information for the `SUNWman` package installed on your system.

```
# pkginfo -l SUNWman
```

What is listed for the status of, the install date of, the number of files used by, and the number of blocks used by this package?

Status:	*Completely installed*
Install date:	*Should match the date and time when you installed Solaris OE on your system*
Number of files:	*xxxx installed path names, x shared directories, xx directories*
Number of blocks:	*xxxxx*

4. Display a long-format listing of the information for the `SUNWman` package on the Solaris 9 OE Software 2 of 2 CD-ROM. Obtain the same information as in the previous step.

```
# pkginfo -d /cdrom/cdrom0/Solaris_9/Product -l SUNWman
```

Status:	*Spooled*
Install date:	*No install date indicated*
Number of files:	*xxxx spooled path names, xx directories, x package information files*
Number of blocks:	*xxxxx*

5. Remove the `SUNWman` package from your system, and verify that it has been removed by trying to access the manual pages.

```
# pkgrm SUNWman
# pkginfo SUNWman
ERROR: information for "SUNWman" was not found
# man ls
No manual entry for ls.
```

6. Reinstall the `SUNWman` package from the Solaris 9 OE Software 2 of 2 CD-ROM. Respond `y` to questions asked by the `pkgadd` command. Verify that the manual pages work.

```
# pkgadd -d /cdrom/cdrom0/Solaris_9/Product SUNWman
# man ls
```

The manual page for `ls` appears.

7. Remove the `SUNWdoc` package from your system.

```
# pkgrm SUNWdoc
```

8. Answer yes to questions from the `pkgrm` command.
9. Are there any package dependencies related to removing this package?

Yes there are. They are six other packages related to the `SUNWdoc` package.

10. Eject the Solaris 9 Software 2 of 2 CD-ROM, and insert the Solaris 9 Software 1 of 2 CD-ROM. Use the `pkgadd` command to spool the `SUNWdoc` package into the default spool area.

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product -s spool SUNWdoc
```

11. Use the `pkginfo` command with the appropriate options to verify the presence of the `SUNWdoc` package in the default spool area.

```
# pkginfo -d spool SUNWdoc
system   SUNWdoc
# pkginfo -d /var/spool/pkg -l SUNWdoc
PKGINST:   SUNWdoc
(further output ommited)
```

12. Install the `SUNWdoc` package. Observe the messages, and verify that the package is installed from the `/var/spool/pkg` directory.

```
# pkgadd SUNWdoc
Processing package instance <SUNWdoc> from
</var/spool/pkg>
(further output omitted)
```

13. Remove the `SUNWdoc` package from the default spool area.

```
# pkgrm -s spool SUNWdoc
```

14. Verify that the `SUNWdoc` package no longer exists in the spool area and that it is installed on your system.

```
# pkginfo -d spool SUNWdoc
ERROR: information for "SUNWdoc" was not found
# pkginfo -l SUNWdoc
PKGINST:   SUNWdoc
(further output omitted)
```

15. Eject the Solaris 9 OE Software 1 of 2 CD-ROM.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 7

Managing Software Patches on the Solaris 9 OE

Objectives

Upon completion of this module, you should be able to:

- Describe the fundamentals of patch administration
- Install and remove patches

The following course map shows how this module fits into the current instructional goal.

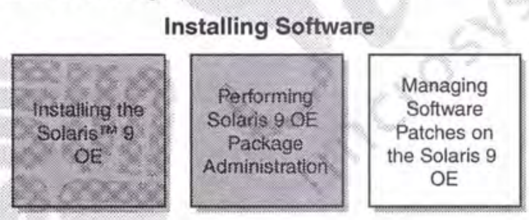

Figure 7-1 Course Map

Preparing for Patch Administration

The administration of patches involves installing or removing Solaris OE patches from a running Solaris OE.

Introducing Solaris OE Patches

A patch contains a collection of files and directories. This collection replaces existing files and directories that prevent proper execution of the software. Some patches contain product enhancements.

The Solaris OE patch types include:

- Standard patches – Patches that fix specific problems with the Solaris OE and other Sun hardware and software products.
- Recommended patches – Solaris OE patches that fix problems that might occur on a large percentage of systems.
- Y2K patches – Patches that ensure compliance of Sun products with the year 2000.
- Patch clusters – A group of standard, recommended, security, or Y2K patches that have been bundled into a single archive for easy downloading and installation.
- Maintenance update – A set of patches that have been tested together and packaged for one-step installation. Available to service contract customers, maintenance updates are designed to update the Solaris OE to a known, tested patch-level. The installation procedure takes much less time than installing the same patches individually.

A patch is distributed as a directory that is identified by a unique number. The number assigned to a patch includes the patch base code first, a hyphen, and a number that represents the patch revision number. For example, a patch directory named `105050-01`, indicates that `105050` is the base code and `01` is the revision number.

The Solaris 9 OE patches are in `zip` format, for example, `105050-01.zip`.

Accessing Patch Documents

Prior to installing patches on your system, you should review the patch documents available through the World Wide Web, patch update CD-ROMs, or anonymous FTP.

To access patch documents through the World Wide Web, go to:

`http://sunsolve.sun.com`

Click Worldwide for a list of alternative sites by geographic areas.

Anonymous FTP access to patch documents is available from `sunsolve.sun.com`. Use your complete email address as a password. After the connection is complete, the publicly available patch documents are located in the `/pub/patches` directory.

Table 7-1 shows important summary documents that list all recommended patches for the Solaris OE.

Table 7-1 Patch Documents and Files

Patch Document	Contents
`Solaris9.PatchReport`	A summary of all patches for the Solaris 9 OE release
`9_Recommended.README`	Instructions for how to install the recommended patch cluster for the Solaris 9 OE, as well as any important notes or warnings, special installation instructions, and usually a note to reboot the system

When you are reviewing patch documentation, start with the Patch Report document first. This report is divided into categories that include information about all patches for a Solaris OE release.

Note – Not all patches available from Sun Microsystems must be installed. Only install the recommended patches, security patches, and those required to fix problems specific to your site.

Checking Patch Levels

Before installing operating environment patches, you should know about patches that have been previously installed on a system.

The showrev command and the patchadd command provide useful information about currently installed patches.

```
# showrev -p
Patch: 106793-01 Obsoletes:  Requires:  Incompatibles: Packages: SUNWhea
. . .
# patchadd -p
Patch: 106793-01 Obsoletes: Requires: Incompatibles: Packages: SUNWhea
. . .
```

Note – Command output is the same for the patchadd -p and showrev -p commands; however, the patchadd command takes longer to display patch information. The showrev command is a binary, and the patchadd command is a script.

Historical information about all patches that are currently installed on a system and that can be uninstalled using the patchrm command is stored in the /var/sadm/patch directory.

The following command lists the contents of the /var/sadm/patch directory.

```
# ls /var/sadm/patch
107558-05  107594-04  107630-01  107663-01  107683-01 107696-01
107817-01  107582-01  107612-06  107640-03
```

Caution – Deleting files from the /var/sadm directory to make more space is a Solution Center call generator. The only way to correct the problems that occur is to restore the deleted files from backup tapes or to reload the software.

Note – It is important to ensure that sufficient space has been allocated for the /var file system. Be sure that the /var file system is at least 200 Mbytes in size to start with. There must be sufficient space for the /var/sadm directory to grow as new software packages and patches are installed on the system.

Obtaining Patches

Sun customers who have a maintenance contract have access to the SunSolve[SM] program's database of patches and patch information, technical white papers, the Symptom and Resolution database, and more. These are available using the World Wide Web or anonymous FTP.

A SunService[SM] program customer can request to receive the patch update CD-ROMs, which are released quarterly.

Sun customers without maintenance contracts have access to a subset of the patches available through the SunSolve program. These patches are available at no charge and include important security and bug fix patches.

To access patches through the World Wide Web, use the following Universal Resource Locators (URLs):

`http://sunsolve.sun.com` – United States

`http://sunsolve.sun.com.au` – Australia

`http://sunsolve.sun.fr` – France

`http://sunsolve.sun.de` – Germany

`http://sunsolve.sun.co.jp` – Japan

`http://sunsolve.sun.se` – Sweden

`http://sunsolve.sun.ch` – Switzerland

`http://sunsolve.sun.co.uk` – United Kingdom

The comprehensive set of patches and patch information is available to contract customers through the button labeled Login. The customer's assigned SunService program password is required to access this database.

To access patches using FTP, use the `ftp` command to connect to:

`sunsolve.sun.com`

The `ftp` utility has many commands; however, only a few are necessary for moving files from system to system. You can locate and copy patches to the local system with a few basic FTP commands.

The following example shows the procedure for changing to the /var/tmp directory on the local system, connecting to the remote FTP site, locating a patch and its README file in the /pub/patches directory, and transferring both files to the local system's directory.

Note – The default mode for an ftp connection is binary mode in Solaris 9 OE. The default mode for an ftp connection in Solaris 8 or earlier versions is American Standard Code for Information Interchange (ASCII) mode. You use the bin command to set the FTP transfer mode to binary mode to transfer binary, image, or a non-text files in these earlier versions of the OE

```
# cd /var/tmp
# ftp sunsolve.sun.com
Connected to sunsolve.sun.com.
220-
220-Welcome to the SunSolve Online FTP server.
220-
220-Public users may log in as anonymous.
220-
220-Contract customers should use the following 2-tier login procedure:
220-
220-At the 1st login prompt: sunsolve
220-             passwd: sunmicro
220-
220-At the 2nd login prompt: <sunsolve login name>/<sunsolve passwd>
220-example:  myssID/mypasswd
220-
220-Public users may log in as anonymous; contract customers
220-should use the standard sunsolve login and password,
220-followed by their SunSolve account/password when prompted.
220-
220-
220 sunsolve8 FTP server (Version wu-2.6.2(21) Thu Mar 14 14:48:19 MST
2002) ready.
Name (sunsolve:usera): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:yourpassword
230-Please read the file README
230-  it was last modified on Mon Aug 26 15:27:12 2002 - 113 days ago
230 Guest login ok, access restrictions apply.
ftp> cd /pub/patches
ftp> get Solaris9.PatchReport
200 PORT command successful.
```

```
150 Opening ASCII mode data connection for Solaris9.PatchReport (8187
bytes).
226 Transfer complete.
local: Solaris9.PatchReport remote: Solaris9.PatchReport
8432 bytes received in 4.9 seconds (1.7 Kbytes/s)
ftp> mget 112605*
mget 112605-01.zip? y
200 PORT command successful.
150 Opening ASCII mode data connection for 112605-01.zip (95744 bytes).
226 Transfer complete.
local: 112605-01.zip remote: 112605-01.zip
96052 bytes received in 1 seconds (91 Kbytes/s)
mget 112605.readme? y
200 PORT command successful.
150 Opening ASCII mode data connection for 112605.readme (2198 bytes).
226 Transfer complete.
local: 112605.readme remote: 112605.readme
2274 bytes received in 0.039 seconds (57 Kbytes/s)
ftp> bye
```

Note – To disable interactive prompting during multiple (`mget`) file transfers, you can begin a session using `ftp -i` *`sitename`* or use the `prompt` command at the `ftp>` prompt.

Preparing Patches for Installation

When patches are downloaded to the local system, you must place the patches in a temporary directory to prepare them for installation. The directory commonly used is the `/var/tmp` directory.

One of the common reasons for patch installation failure is directory permission or ownership problems. The `/var/tmp` directory is open to all and eliminates any of these types of problems.

The Solaris 7, Solaris 8, and Solaris 9 OE patches are in `zip` format, for example, `105050-01.zip`.

Use the `unzip` command to unpack the patch files.

```
# /usr/bin/unzip 105050-01.zip
```

Earlier versions of the Solaris OE used compressed `tar` files in a `tar.Z` format, for example, `101010-01.tar.Z`

Use the `zcat` command to uncompress the patch files and the `tar` command to create the patch directories.

```
# /usr/bin/zcat 105050-01.tar.Z | tar xvf -
```

Patch Contents

Figure 7-2 shows the contents of a patch directory after it is extracted from the `zip` file.

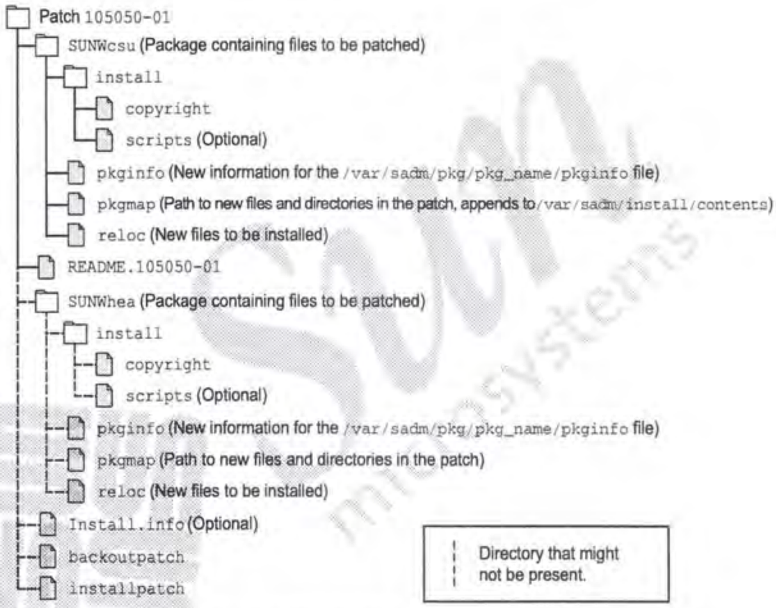

Figure 7-2 An Extracted Patch Directory

Installing and Removing Patches

The two most common commands for managing patches are:

- `patchadd` – Installs uncompressed patches to the Solaris OE
- `patchrm` – Removes patches installed on the Solaris OE

Additionally, you install cluster patches by using the `install_cluster` command. You can also manage patches through the Solaris Management Console.

Installing a Patch

When a patch is installed, the `patchadd` command calls the `pkgadd` command to install the patch packages.

The following example shows the procedure for patch installation. This example assumes that the patch to be installed exists in the `/var/tmp` directory and has been unzipped or uncompressed for installation.

```
# cd /var/tmp
# patchadd 105050-01
Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)
Installing patch packages...
Patch number 105050-01 has been successfully installed.
See /var/sadm/patch/105050-01/log for details.
Patch packages installed:
  SUNWhea
```

Figure 7-3 shows those components of the `/var/sadm` directory that are updated during the installation of patch `105050-01`.

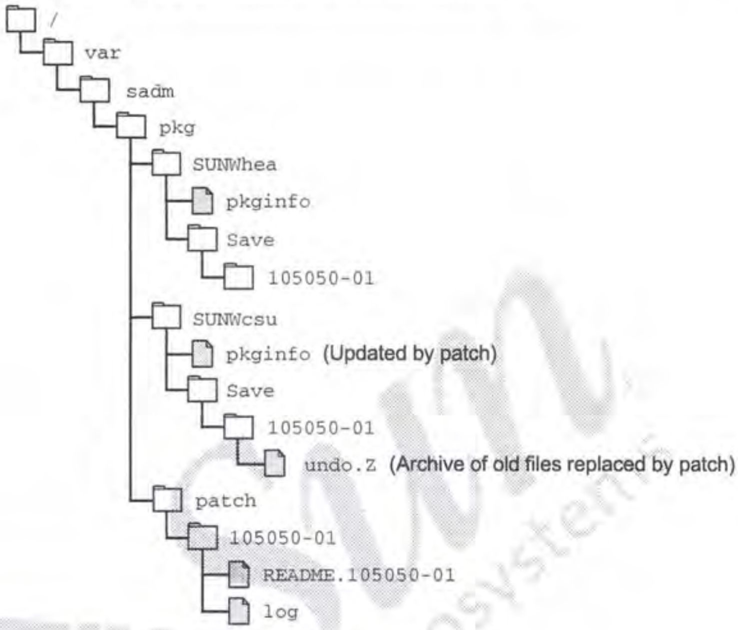

Figure 7-3 Updated `/var/sadm` Directories

Error codes can be useful for troubleshooting problems encountered during installation.

Some of the error codes found in the `/usr/sbin/patchadd` command are shown in Table 7-2.

Table 7-2 Some `/usr/sbin/patchadd` Command Error Codes

Number	**`patchadd`** Error Codes
0	No error.
1	Usage error.
2	An attempt to apply a patch that has already been applied.

Table 7-2 Some /usr/sbin/patchadd Command Error Codes (Continued)

Number	**patchadd** Error Codes
3	The effective user ID (EUID) is not root.
4	An attempt to save original files failed.
5	The pkgadd command failed.
6	The patch is obsolete.
7	An invalid package directory.
8	An attempt to patch a package that is not installed.
9	Cannot access /usr/sbin/pkgadd (client problem).
10	Package validation errors.
11	An error occurred while adding a patch to the root template.
12	The patch script terminated due to a signal.
13	A symbolic link was included in the patch.
14	Not used.
15	The prepatch script had a return code other than 0.
16	The postpatch script had a return code other than 0.
17	A mismatch of the -d option occurred between a previous patch installation and the current one.
18	There is not enough space in the file systems that are targets of the patch.
19	The $SOFTINFO/INST_RELEASE file was not found.
20	A direct instance patch was required but was not found.
21	The required patches have not been installed on the manager.
22	A progressive instance patch was required but was not found.
23	A restricted patch is already applied to the package.
24	An incompatible patch was applied.
25	A required patch was not applied.

Table 7-2 Some `/usr/sbin/patchadd` Command Error Codes (Continued)

Number	**`patchadd` Error Codes**
26	The user-specified backout data cannot be found.
27	The relative directory supplied cannot be found.
28	A `pkginfo` file is corrupt or missing.
29	Bad patch ID format.
30	Dry run failures occurred.
31	The path given for the `-C` option was invalid.
32	You must be running the Solaris 2.6 OE to the Solaris 9 OE.
33	The patch file was formatted incorrectly or the patch file was not found.
34	An incorrect patch spool directory was given.
35	A later revision was already installed.
36	You cannot create a safe temporary directory.
37	An illegal backout directory was specified.

Removing a Patch

When you remove a patch, the `patchrm` command restores all files that were modified or replaced by that patch, unless:

- The patch was installed with the `patchadd -d` option (which instructs the `patchadd` command not to save copies of files being updated or replaced)
- The patch is required by another patch
- The patch has been obsoleted by a later patch

The `patchrm` command calls the `pkgadd` utility to restore packages that were saved during the initial patch installation.

The following example shows how to remove a patch by using the `patchrm` command.

```
# patchrm 105050-01
Checking installed packages and patches...
Backing out patch 105050-01...
Patch 105050-01 has been backed out.
#
```

Installing Patch Clusters

The patch cluster provides a selected set of patches for a designated Solaris OE level and is conveniently wrapped for one-step installation. Patch clusters are usually a set of recommended, security, or Y2K patches.

You should not install cluster patches on systems with limited disk space. Sun does not recommend installing cluster patches on systems with less than 10 Mbytes of available space in the `/`(root), `/usr`, `/var`, or `/opt` partitions.

By default, the cluster installation procedure saves the base objects being patched. Prior to installing the patches, the cluster installation script first determines if enough system disk space is available in the `/var/sadm/pkg` directory to save the base packages and terminates if not enough space is available.

You can override the save feature by using the `-nosave` option when you are executing the cluster installation script. If you use the `-nosave` option, you will not be able to back out individual patches if the need arises.

You can remove individual patches that were installed by the patch cluster by using the `patchrm` command. The `README` file is located in the specific patch directory under the `/var/sadm/patch` directory after the patch has been installed.

To install a patch cluster, perform the following steps:

1. Be sure the patch cluster has been unzipped or uncompressed and extracted if the cluster was received as a `tar.Z` file.
2. Decide on which method to use to install the cluster—the recommended default `save` option or the `-nosave` option.
3. Change to the directory that contains the patch cluster. Read the `CLUSTER_README` file, which contains information about the bundled set of patches, including:
   - Cluster description
   - Patches included
   - Important notes and warnings
   - Save and backout options
   - Special install instructions
   - Special patch circumstances
   - Any notices and other recommendations

   Then run the `install_cluster` script.

```
# cd 9_Recommended
# ./install_cluster
```

The installation appears as follows:

```
# ./install_cluster

  Patch cluster install script for Solaris 9 Recommended

*WARNING* SYSTEMS WITH LIMITED DISK SPACE SHOULD *NOT* INSTALL PATCHES:
(Other disk space warning messages omitted)

Are you ready to continue with install? [y/n]:y
Determining if sufficient save space exists...
Sufficient save space exists, continuing...
Installing patches located in /tmp/9_Recommended
Using patch_order file for patch installation sequence
Installing 113319-01...
(other patch messages omitted)

The following patches were not able to be installed:
        112875-01
        113023-01

For more installation messages refer to the installation logfile:
  /var/sadm/install_data/Solaris_9_Recommended_log
```

```
Use '/usr/bin/showrev -p' to verify installed patch-ids.
Refer to individual patch README files for more patch detail.
Rebooting the system is usually necessary after installation
#
```

4. Read each individual patch README file to determine if any additional steps are required to fully install any individual patch.
5. Check the log file if more detail is needed.

   Reviewing the log provides information about why the patches listed above were not able to be installed:

```
# more /var/sadm/install_data/Solaris_9_Recommended_log
*** Install Solaris 9 Recommended begins Tue Dec 17 14:47:11 MST 2002 ***
*** PATCHDIR = /tmp/9_Recommended ***
(output omitted)
Installing 112875-01...

Checking installed patches...
Patch 112875-01 has already been applied.
See patchadd(1M) for instructions.

Installing 113023-01...

Checking installed patches...
One or more patch packages included in
113023-01 are not installed on this system.
(output omitted)
#
```

6. Reboot the system for all patches to take effect.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Maintaining Patches (Level 1)

In this exercise, you transfer a patch from a classroom server, apply the patch, and then remove it.

Preparation

Your instructor should provide directions for accessing a patch on a server that is available to systems in the classroom. Refer to the lecture notes as necessary to perform the tasks listed.

Tasks

Complete the following tasks:

- Create a directory to hold patches. Use the `ftp` command to transfer a patch from a classroom server into the directory you create. Unzip the patch. Verify that no patch has been applied to your system. Verify that the `/var/sadm/patch` directory is empty.
- Read the `README` file associated with the patch to verify which Solaris OE release is appropriate for the patch. Add the patch, and verify that it is installed in the `/var/sadm/patch` directory. View the log file for this patch.
- Remove the patch you just installed, and verify that it is no longer applied to the system.

Exercise: Maintaining Patches (Level 2)

In this exercise, you transfer a patch from a classroom server, apply the patch, and then remove it.

Preparation

Your instructor should provide directions for accessing a patch on a server that is available to systems in the classroom. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Create a directory to hold patches. Use the `ftp` command to transfer a patch from a classroom server into the directory you create. Unzip the patch. Verify that no patch has been applied to your system. Verify that the `/var/sadm/patch` directory is empty.
- Read the `README` file associated with the patch to verify which Solaris OE release is appropriate for the patch. Add the patch, and verify that it is installed in the `/var/sadm/patch` directory. View the log file for this patch.
- Remove the patch you just installed, and verify that it is no longer applied to the system.

Tasks

Complete the following steps:

1. Create a directory to hold patches. Use the binary transfer mode of the `ftp` command to transfer a patch from a classroom server into the directory you created. Your instructor should provide information about where to find a patch on the server. Close your `ftp` connection when you are finished.

Note – The default mode for an `ftp` connection is binary mode in Solaris 9 OE. The default mode for an `ftp` connection in Solaris 8 or earlier versions is ASCII mode. You use the `bin` command to set the FTP transfer mode to binary mode to transfer binary, image, or an non-text files in these earlier versions of the OE.

---

2. Use the `unzip` command to extract the patch from the zip archive.
3. Use the `patchadd` command to determine if any patches are currently installed on your system.
4. Verify that the `/var/sadm/patch` directory is empty.
5. Read the `README` file that is associated with the patch you unzipped. Verify the Solaris OE release for which the patch is required.

   Solaris OE release:
6. Add the patch.
7. Verify that the patch is installed. What are the packages that the patch affects?
8. Examine the patch installation log.
9. Remove the patch you just installed. Verify that the patch is no longer installed.

Exercise: Maintaining Patches (Level 3)

In this exercise, you transfer a patch from a classroom server, apply the patch, and then remove it.

Preparation

Your instructor should provide directions for accessing a patch on a server that is available to systems in the classroom. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

In this exercise, you accomplish the following:

- Create a directory to hold patches. Use the `ftp` command to transfer a patch from a classroom server into the directory you create. Unzip the patch. Verify that no patch has been applied to your system. Verify that the `/var/sadm/patch` directory is empty.
- Read the `README` file associated with the patch to verify which Solaris OE release is appropriate for the patch. Add the patch, and verify that it is installed in the `/var/sadm/patch` directory. View the log file for this patch.
- Remove the patch you just installed, and verify that it is no longer applied to the system.

Tasks and Solutions

Complete the following steps:

Note – The default mode for an `ftp` connection is binary mode in Solaris 9 OE. The default mode for an `ftp` connection in Solaris 8 or earlier versions is ASCII mode. You use the `bin` command to set the FTP transfer mode to binary mode to transfer binary, image, or an non-text files in these earlier versions of the OE.

---

1. Create a directory to hold patches. Use the binary transfer mode of the `ftp` command to transfer a patch from a classroom server into the directory you created. Your instructor should provide information about where to find a patch on the server. Close your `ftp` connection when you are finished. For example:

```
# cd /var/tmp
# ftp instructor
(connection and login messages)
ftp> cd /export/patches
ftp> get 112875-01.zip
(ftp messages)
ftp> bye
221 Goodbye.
#
```

2. Use the `unzip` command to extract the patch from the zip archive, for example:

```
# unzip 112875-01.zip
Archive:  112875-01.zip
   creating: 112875-01/
  inflating: 112875-01/.diPatch
  inflating: 112875-01/patchinfo
   creating: 112875-01/SUNWrcmds/
  inflating: 112875-01/SUNWrcmds/pkgmap
  inflating: 112875-01/SUNWrcmds/pkginfo
   creating: 112875-01/SUNWrcmds/install/
  inflating: 112875-01/SUNWrcmds/install/checkinstall
  inflating: 112875-01/SUNWrcmds/install/copyright
  inflating: 112875-01/SUNWrcmds/install/i.none
  inflating: 112875-01/SUNWrcmds/install/patch_checkinstall
  inflating: 112875-01/SUNWrcmds/install/patch_postinstall
  inflating: 112875-01/SUNWrcmds/install/postinstall
  inflating: 112875-01/SUNWrcmds/install/preinstall
   creating: 112875-01/SUNWrcmds/reloc/
```

```
   creating: 112875-01/SUNWrcmds/reloc/usr/
   creating: 112875-01/SUNWrcmds/reloc/usr/lib/
   creating: 112875-01/SUNWrcmds/reloc/usr/lib/netsvc/
   creating: 112875-01/SUNWrcmds/reloc/usr/lib/netsvc/rwall/
  inflating: 112875-01/SUNWrcmds/reloc/usr/lib/netsvc/rwall/rpc.rwalld
  inflating: 112875-01/README.112875-01
#
```

3. Use the patchadd command to determine if any patches are currently installed on your system.

```
# patchadd -p
```

The patchadd *command should display a message.*

4. Verify that the /var/sadm/patch directory is empty.

```
# ls /var/sadm/patch
#
```

5. Read the README file that is associated with the patch you unzipped. Verify the Solaris OE release for which the patch is required.

```
# more 112875-01/README.112875-01
Patch-ID# 112875-01
Keywords: security rpc.rwalld string
Synopsis: SunOS 5.9: patch /usr/lib/netsvc/rwall/rpc.rwalld
Date: Jun/21/2002
(output omitted)
```

6. Add the patch.

```
# patchadd 112875-01
Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...
Patch number 112875-01 has been successfully installed.
See /var/sadm/patch/112875-01/log for details
Patch packages installed:
  SUNWrcmds
#
```

7. Verify that the patch is installed. What are the packages that the patch affects?

```
# patchadd -p
Patch: 112875-01 Obsoletes: Requires: Incompatibles: Packages: SUNWrcmds
```

8. Examine the patch installation log file.

```
# cd /var/sadm/patch/112875-01
# more log
(output omitted)
Installation of <SUNWrcmds> was successful.
```

9. Remove the patch you just installed. Verify that the patch is no longer installed.

```
# cd
# patchrm 112875-01
Checking installed patches...
Backing out patch 112875-01...
Patch 112875-01 has been backed out.
# patchadd -p
```

The `patchadd -p` command should not contain any reference to 112875-01.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 8

Executing Boot PROM Commands

Objectives

Upon completion of this module, you should be able to:

- Identify boot programmable read-only memory (PROM) fundamentals
- Use basic boot PROM commands
- Identify the system's boot device
- Create and remove custom device aliases
- View and change nonvolatile random access memory (NVRAM) parameters from the shell
- Interrupt an unresponsive system

The following course map shows how this module fits into the current instructional goal.

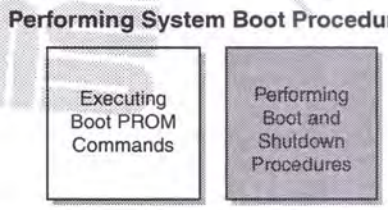

Figure 8-1 Course Map

Introducing Boot PROM Fundamentals

All Sun systems have resident boot PROM firmware that provides basic hardware testing and initialization prior to booting. The boot PROM also enables you to boot from a wide range of devices. In addition, there is a user interface that provides several important functions.

The Sun boot PROM has access to a standard set of generic device drivers. The system needs these drivers to access and control the buses and the boot device to boot the system properly.

All versions of the OpenBoot™ architecture allow a third-party board to identify itself and load its own plug-in device driver. Each device identifies its type and furnishes its plug-in device driver when requested by the OpenBoot PROM during the system hardware configuration phase of the boot process.

Figure 8-2 shows the identification process.

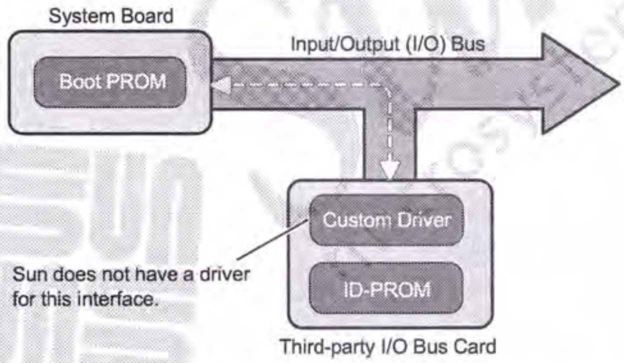

Figure 8-2 Third-Party Device Identification Process

Goal of the OpenBoot™ Architecture Standard

The overall goal of the Institute of Electrical and Electronics Engineers (IEEE) standard for the OpenBoot architecture is to provide the capabilities to:

- Test and initialize system hardware
- Determine the system's hardware configuration
- Boot the operating environment
- Provide an interactive interface for configuration, testing, and debugging
- Enable the use of third-party devices

Boot PROM

Each Sun system has a boot PROM chip. This 1-Mbyte chip is typically located on the same board as the central processing unit (CPU). Boot PROM chips are usually found in a pluggable socket on older systems. As of the 3.*x* PROM, they are permanently soldered to the main system board.

The Ultra™ workstations use a reprogrammable boot PROM called a flash PROM (FPROM). The FPROM allows you to load new boot program data into the PROM by using software, instead of having to replace the chip. A CD-ROM distributes these updates.

Desktop systems have a write-protect jumper that must be moved before you can write to the PROM. You have to move the jumper because the default position is write-protect. Refer to the *Flash Programming Manual for Ultra Desktop Systems*, part number 802-3233-17, for the jumper location on your system.

Caution – Many systems have the jumper under an installed frame buffer or other removable card. Be careful when removing or replacing this card.

The main functions of the boot PROM are to test the system hardware and to boot the operating environment. The boot PROM firmware is referred to as the *monitor* program.

The boot PROM firmware controls the operation of the system before the operating environment has been booted and the `kernel` is available. The boot PROM also provides the user with a user interface and firmware utility commands, known as the FORTH command set. Commands include the boot commands, diagnostics commands, and commands to modify the default configuration.

Note – The boot PROM does not understand the Solaris™ Operating Environment (Solaris OE) file systems or files. It handles mainly hardware devices.

Currently, there are five generations of Sun boot PROMs. Each generation has its own base revision number, as described in the following list:

- 1.*x* — The first boot PROM used on SPARC® systems.
- 2.*x* — The first OpenBoot PROM, updated by replacing the application-specific integrated circuits (ASICs).
- 3.*x* — The OpenBoot PROM non-removable ASIC, updated by flash. It was the first 64-bit capable PROM. This PROM was introduced with the Ultra 1 workstation and used with UltraSPARC I and UltraSPARC II workstations and servers.
- 4.*x* — Used in Sun Blade™ and Sun Fire™ entry-level and mid-range workstations and servers. Used in UltraSPARC® IIi and UltraSPARC III workstations and servers, such as Sun Blade 150 and 2000 workstations and Sun Fire 280R, 480R, v880z servers.
- 5.*x* — Used in mid-range and high-end servers using UltraSPARC III microprocessors, such as the Sun Fire 3800, Sun Fire 4800, Sun Fire 4810, Sun Fire 6800, Sun Fire 12K and Sun Fire 15K servers.

To determine which revision of OpenBoot PROM is running on the system, perform the command:

```
# /usr/platform/`uname -m`/sbin/prtdiag -v
```

NVRAM

Another important hardware element in each Sun system is the NVRAM chip. This removable chip is often located on the main system board.

The NVRAM module contains the electronically erasable programmable read-only memory (EEPROM). The EEPROM stores user-configurable parameters that have been changed or customized from the boot PROM's default parameters settings. This behavior gives you a certain level of flexibility in configuring the system to behave in a particular manner for a specific set of circumstances. A single lithium battery within the NVRAM module provides battery backup for the NVRAM and the clock.

The NVRAM contains editable and noneditable areas. The noneditable area includes the following:

- The Ethernet address, such as `8:0:20:5d:6f:9e`
- The system host ID value, such as `805d6f9e`

The editable area includes the following:

- The time-of-day (TOD) clock value
- The configuration data describing system operating parameters
- A diagnostic mode switch that enables or disables power-on self-test (POST)
- The device name and the path to the default boot device
- A location for customized programming that is used during the boot process

---

Note – Remember to retain the NVRAM chip, because it contains the host ID. Many licensed software packages are based on the system host ID. The NVRAM chip has a yellow sticker with a bar code on it. If the chip fails, Sun can replace the chip if given the numbers below this bar code. The replacement chip has the same host ID and Ethernet address. It can be plugged into the same location on the motherboard as the chip it is replacing.

Figure 8-3 shows the basic elements of the Boot PROM and NVRAM.

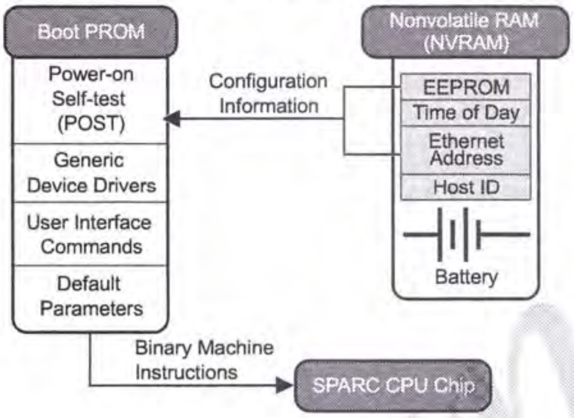

Figure 8-3 Basic Elements of the Boot PROM and NVRAM

POST

When a system's power is turned on, a low-level POST is initiated. This low-level POST code is stored in the boot PROM and is designed to test the most basic functions of the system hardware.

At the successful completion of the low-level POST phase, the boot PROM firmware takes control and performs the following initialization sequence:

- Probes the memory and then the CPU
- Probes bus devices, interprets their drivers, and builds a device tree
- Installs the console

After the boot PROM initializes the system, the banner displays on the console. The system checks parameters stored in the boot PROM and NVRAM to determine if and how to boot the operating environment.

Controlling the POST Phase

One of the first tests that POST runs is to check to determine if a keyboard is connected to the system and if a Stop-key option is present.

Note – You can control the POST phase through the Sun keyboard only.

The Stop key is located on the left side of the keyboard. To enable various diagnostic modes, hold down the Stop key simultaneously with another key. The Stop-key sequences have an effect on the OpenBoot PROM and define how POST runs when a system's power is turned on. The following is a list of the Stop-key sequences:

- Stop-D key sequence – Hold down the Stop and D keys simultaneously while system power is turned on, and the firmware automatically switches to diagnostic mode. This mode runs more extensive POST diagnostics on the system hardware. The OpenBoot PROM variable `diag-switch?` is set to `true`.

See Figure 8-4 on page 8-7 to show the effect of the variable `diag-switch?`.

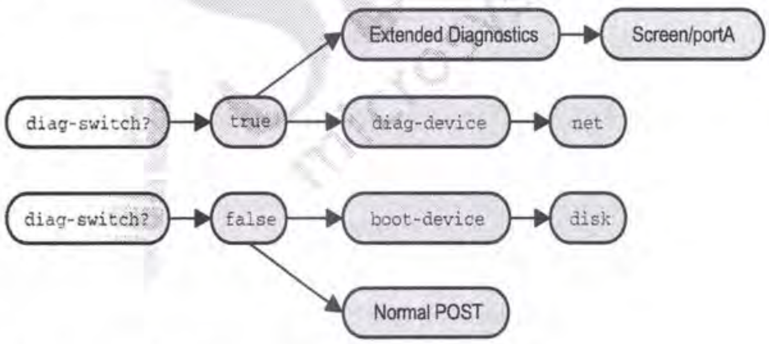

Figure 8-4 Post Diagnostics

Note – The Stop-D key sequence is not available on a serial port terminal.

- Stop-N key sequence – Hold down the Stop and N keys simultaneously while the system power is turned on to set the NVRAM parameters to the default values. When you see the light emitting diodes (LEDs) on the keyboard begin to flash, you can release the keys, and the system should continue to boot.

  Incorrect NVRAM settings can cause system boot failure. For example, during a flash PROM download, if a power failure occurs, some of the contents of the NVRAM can become unusable.

  If the system does not boot and you suspect that the NVRAM parameters are set incorrectly, the parameters can easily be changed to the default values.

- Stop-A key sequence – Hold down the Stop and A keys simultaneously to interrupt any program that is running at the time these keys are pressed and to put the system into the command entry mode for the OpenBoot PROM. The system presents an `ok` prompt for the user, which signifies it is ready to accept OpenBoot PROM commands.

Caution – The Stop-A key sequence, as a method for getting to the `ok` prompt, is not recommended unless there is absolutely no alternative. The Stop-A key sequence can cause Solaris OE file system corruption which can be difficult to repair.

Disabling the Abort Sequence

As a system administrator, you might want to disable the abort key sequence on a system to prevent possible corruption of a file system or to provide tighter security.

To disable the abort key sequence, edit the `/etc/default/kbd` file. Inside the file, the statement `KEYBOARD_ABORT=disable` is commented out. Remove the comment from in front of the value, save the file, and execute the command `kbd -i`. When you have completed these steps, the system allows Stop-A key sequence only during the boot process.

You can also configure the system to change the keyboard abort sequence to an alternate keystroke. Review the man page for the `kbd` command for more information.

Displaying POST to the Serial Port

As the system administrator, you can attach a terminal to the serial port of a system to capture a far greater amount of information from the POST output.

When the power is turned on, POST looks for a keyboard. If there is no keyboard present, POST diverts system output to serial port A.

POST runs more extensive tests when the system is in diagnostic mode with the PROM parameter `diag-switch?` set to `true`.

Be sure to attach the correct type of null modem cable for your system type to serial port A.

Some systems require a special adapter cable. Connect the other end of the cable to the modem port of the American Standard Code for Information Interchange (ASCII) terminal, as shown in Figure 8-5.

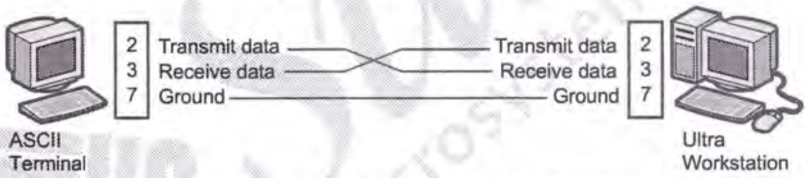

Figure 8-5 Serial Port Connection to a Sun Workstation

The following example is the POST output from a Sun Blade™ 1000 workstation:

```
Partial Post Output Listing
@(#) 4.0 Version 29 created 2000/07/12 16:46
Clearing TLBs Done
Reset: 0000.0000.0000.0010 SPOR
Loading Configuration
Membase: 0000.0000.0000.0000
MemSize: 0000.0000.2000.0000
Init CPU arrays Done
Init E$ tags Done
Setup TLB Done
MMUs ON
Block Scrubbing Done
Copy Done
PC = 0000.07ff.f000.3138
Decompressing Done
```

```
Size = 0000.0000.0006.e3b0
ttya initialized
Start Reason: Soft Reset
System Reset: (SPOR)
Probing gptwo at 0,0 SUNW,UltraSPARC-III (750 MHz @ 5:1, 8 MB)
   memory-controller
Probing gptwo at 1,0 Nothing there
Probing gptwo at 8,0 pci pci upa ppm
Loading Support Packages: kbd-translator
Loading onboard drivers: ebus flashprom bbc ppm i2c dimm-fru dimm-fru
   dimm-fru dimm-fru dimm-fru dimm-fru dimm-fru dimm-fru nvram idprom
   i2c cpu-fru temperature fan-control card-reader motherboard-fru
Memory Configuration:
Segment @ Base:        0  Size:   512 MB (2-Way)
Probing /upa@8,480000 Device 0,0 Nothing there
Probing /upa@8,480000 Device 1,0 Nothing there
Probing /pci@8,600000 Device 4  SUNW,qlc fp disk
Probing /pci@8,600000 Device 1  Nothing there
Probing /pci@8,700000 Device 5  network firewire usb
dev-descrip
next-add
node made
Probing /pci@8,700000 Device 6  scsi disk tape scsi disk tape
Probing /pci@8,700000 Device 1  Nothing there
Probing /pci@8,700000 Device 2  Nothing there

 (UltraSPARC-III) , Keyboard Present
OpenBoot 4.0, 512 MB memory installed, Serial #12134217.
Ethernet address 8:0:20:b9:27:49, Host ID: 80b92749.
```

Using Basic Boot PROM Commands

The boot PROM monitor provides a user interface for invoking OpenBoot commands.

Note – The `ok` prompt indicates that the Solaris OE is currently not running.

Table 8-1 shows some of the commands typically entered at the `ok` prompt.

Table 8-1 Typical Commands Used at the `ok` Prompt

Command	Description
`banner`	Displays the power-on banner
`boot`	Boots the system
`help`	Lists the main help categories
`printenv`	Displays all parameters' current and default values
`setenv`	Sets the specified NVRAM parameter to some value
`reset-all`	Resets the entire system; similar to a power cycle
`set-defaults`	Resets all parameter values to the factory defaults
`sifting` *`text`*	Displays the FORTH commands containing *`text`*
`.registers`	Displays the contents of the registers
`probe-scsi`	Identifies the devices on the internal Small Computer System Interface (SCSI) bus
`probe-scsi-all`	Identifies the devices on all SCSI buses
`probe-ide`	Identifies devices on the internal integrated device electronics (IDE) bus
`probe-fcal-all`	Identifies devices on all Fibre Channel loops
`show-devs`	Displays the entire device tree
`devalias`	Identifies the current boot device alias for the system

Table 8-1 Typical Commands Used at the ok Prompt (Continued)

nvalias	Creates a new device alias name
nvunalias	Removes a device alias name
show-disks	Displays and allows a selection of device paths for the disks to be used for nvalias
sync	Manually attempts to flush memory and synchronize file systems
test	Runs self-tests on specified devices

Identifying the System Boot PROM Version

The banner command lists useful information about the system, such as the model name, the boot PROM version number (for example, 1.*x*, 2.*x*, 3.*x*, 4.*x*, or 5.*x*), the amount of memory, the Ethernet address, and the host ID.

The following example shows output from the banner command.

```
ok banner
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 360MHz), Keyboard Present
OpenBoot 3.31, 128 MB (50 ns) memory installed, Serial #11888271.
Ethernet address 8:0:20:b5:66:8f, Host ID: 80b5668f.
```

Booting the System

Use the boot command to boot the Solaris OE from the ok prompt. This command has several options available for booting the system in different situations.

The format for the boot command is:

```
ok boot device_name - options
```

Enter the boot command at the ok prompt to boot the system to multiuser mode automatically.

```
ok boot
```

The following list describes the options for the `boot` command:

- `-s` – Boots the system to a single-user mode and asks the user for the `root` password.

```
ok boot -s
ok boot cdrom -s
```

- `-a` – Boots the system interactively. Use this option if an alternative file needs to be executed during boot. The boot program asks for the following information.

```
ok boot -a
Enter filename [kernel/sparcv9/unix]:
Enter default directory for modules [/platform/SUNW,Ultra-5_10/kernel
/platform/sun4u/kernel /kernel /usr/kernel]:
Name of system file [etc/system]:
root file system type [ufs]:
Enter physical name of root device:
```

- `-r` – Performs a reconfiguration boot. Use this option to find a newly attached device and to create new device entries in the `/devices` and `/dev` directories. It also updates the `/etc/path_to_inst` file.

```
ok boot -r
```

- `-v` – Boots the system while displaying more detailed device information to the console. Use this option to troubleshoot problems during the boot process. You can use this option with other options.

```
ok boot -v

ok boot -rv

ok boot -sv
```

Accessing More Detailed Information

You use the help command to obtain help on the main categories in the OpenBoot firmware.

The following is an example of the help output from an Ultra 5 workstation that is running OpenBoot PROM version 3.31:

```
ok help
Enter 'help command-name' or 'help category-name' for more help
(Use ONLY the first word of a category description)
Examples:  help system -or- help nvramrc
Categories:
boot (Load and execute a program)
nvramrc (Store user defined commands)
system configuration variables (NVRAM variables)
command line editing
editor (nvramrc editor)
resume execution
devaliases (Device aliases)
diag (Diagnostics commands)
ioredirect (I/O redirection commands)
misc (Miscellaneous commands)
ok
```

The help command listing provides a number of other keywords that you can use to view further details.

For example, to view specific information for one of the main categories listed in the preceding example, perform one of the following commands:

```
ok help boot

ok help nvramrc

ok help diag

ok help misc
```

Listing NVRAM Parameters

You use the printenv command to list all the NVRAM parameters. If the parameter can be modified, the printenv command displays its default setting and current setting.

The following example shows output from the printenv command.

```
ok printenv
Variable Name           Value                          Default Value

tpe-link-test?          true                           true
scsi-initiator-id       7                              7
keyboard-click?         false                          false
keymap
ttyb-rts-dtr-off        false                          false
ttyb-ignore-cd          true                           true
ttya-rts-dtr-off        false                          false
ttya-ignore-cd          true                           true
ttyb-mode               9600,8,n,1,-                   9600,8,n,1,-
ttya-mode               9600,8,n,1,-                   9600,8,n,1,-
pcia-probe-list         1,2,3,4                        1,2,3,4
pcib-probe-list         1,2,3                          1,2,3
mfg-mode                off                            off
diag-level              max                            max
#power-cycles           273
output-device           screen                         screen
input-device            keyboard                       keyboard
boot-command            boot                           boot
auto-boot?              true                           true
diag-device             net                            net
boot-device             disk net                       disk net
local-mac-address?      false                          false
screen-#columns         80                             80
screen-#rows            34                             34
use-nvramrc?            false                          false
nvramrc                 devalias pgx24 /pci1f,0 ...
security-mode           none
security-password
security-#badlogins     0
diag-switch?            false                          false
ok
```

You can also use the printenv command to display a single parameter and its values.

For example, to display only the `boot-device` parameter, perform the command:

```
ok printenv boot-device
boot-device = disk net
```

The possible values of the `boot-device` parameter include `disk`, `net`, and `cdrom`.

Note – Some OpenBoot PROM parameters, such as `auto-boot?`, end in a question mark. If an OpenBoot PROM parameter ends in a question mark, the parameter value is typically either `true` or `false`.

Changing NVRAM Parameters

You use the `setenv` command to change the current values assigned to NVRAM parameters.

If the `auto-boot?` parameter is set to `true`, the system boots automatically. If it is set to false, the system stops at the `ok` prompt.

The following example changes the `auto-boot?` parameter from its default setting of `true` to the value of `false`.

```
ok printenv auto-boot?
auto-boot? = true
ok
ok setenv auto-boot? false
auto-boot? = false
```

The `reset-all` command halts the system, clears all buffers and registers, and performs a software simulated power-off/power-on of the system.

```
ok reset-all
Resetting ...
```

Note – The `reset-all` command clears system registers, which is required on a system with a PROM 3.*x* or higher before you can use the `probe` command or perform other tests.

Restoring Default NVRAM Parameters

You use the set-defaults command to reset all NVRAM parameters to their default values. It affects only parameters that have assigned default values.

```
ok set-defaults
Setting NVRAM parameters to default values.
ok
```

To reset a specific parameter to its default value, use the set-default command followed by the parameter name.

```
ok set-default parameter-name
```

For example, to reset the diag-level parameter, perform the command:

```
ok set-default diag-level
```

To restore the default NVRAM parameters when you are starting the system, hold down the Stop-N key sequence while the system power is turned on.

When the LEDs on the keyboard begin to flash, release the keys, and the system continues to boot.

Displaying Devices Connected to the Bus

To identify the peripheral devices currently connected to the system, such as disks, tape drives, or CD-ROMs, use the probe command.

To identify the various probe commands that are available with your system, use the sifting command. The sifting command is useful for finding OpenBoot PROM commands when you do not know the exact command syntax.

For example, to find the probe commands available, perform the command:

```
ok sifting probe

(f006c954) probe-all        (f006c5a0) probe-all    (f006c378) probe-ide
(f006c1e8) probe-pci-slot   (f006bc8c) probe-scsi
(f006bd78) probe-scsi-all   (f0060fe8) probe-pci
(output truncated)
```

The most common `probe` commands are the `probe-scsi` command, the `probe-scsi-all` command, and the `probe-ide` command.

Systems that contain the Fibre Channel-Arbitrated Loop (FC-AL) Gigabit Interface Converters (GBICs) use the `probe-fcal-all` command.

Caution – The following warning message might be displayed if you invoke the `probe` commands on Sun systems that contain a boot PROM that is version 3.*x* and above.

```
This command may hang the system if a Stop-A or halt command has been
executed. Please type reset-all to reset the system before executing this
command.
Do you wish to continue? (y/n) n
```

To avoid having your system hang, perform the commands:

```
ok setenv auto-boot? false
ok reset-all
```

One method you can use to tell if the system might hang during a `probe` command is to use the `.registers` command.

```
ok .registers
       Normal       Alternate     MMU       Vector
0:            0             0         0           0
1:            0             0         0           0
2:            0             0         0           0
3:            0             0         0           0
4:            0             0         0           0
(output edited for brevity)
%PC   0   %nPC   0
%TBA  0   %CCR   0  XCC:nzvc    ICC:nzvc
```

The preceding output shows that the registers are all empty, with values of 0 (zero). Should the registers hold values other than 0, the `probe` command would most likely hang the system.

The `probe-scsi` Command

The `probe-scsi` command identifies the peripheral devices attached to the on-board SCSI controller. The `probe-scsi` command identifies such peripheral devices as disks, tape drives, or CD-ROMs by their target addresses.

```
ok probe-scsi
Target 1
Unit 0   Disk     FUJITSU MAB3045S SUN4.2G17059825M62990
Target 3
Unit 0   Disk     IBM     DDRS34560SUN4.2GS98E99255C5917
        (C) Copyright IBM Corp.
         1997. All rights reserved.
Target 6
Unit 0   Removable Read Only device SONY CDROM
```

The `probe-scsi-all` Command

The `probe-scsi-all` command identifies the peripheral devices that are attached to the on-board SCSI controller and all peripheral devices attached to separate SBus or PCI SCSI controllers.

```
ok probe-scsi-all
/pci@1f,0/pci@1/pci@1/SUNW,isptwo@4
Target 3
Unit 0   Disk FUJITSU MAB3045S SUN4.2G1907
Target 4
Unit 0   Removable Tape EXABYTE EXB-8505SMBANSH20090
```

The `probe-ide` Command

The `probe-ide` command identifies disks and CD-ROMs that are attached to the on-board IDE controller. This command displays the device number of the internal device.

```
ok probe-ide
    Device 0          ( Primary Master )
                       ATA Model : ST 38420A (DISK)

    Device 1          ( Primary Slave )
                       Not Present

    Device 2          ( Secondary Master )
                       Removable ATAPI Model : CRD-8322B (CD-ROM)

    Device 3         ( Secondary Slave )
                       Not Present
```

The `probe-fcal-all` Command

The `probe-fcal-all` OpenBoot PROM command identifies peripheral devices on systems containing the FC-AL GBICs. The Sun Enterprise™ 3500 server is an example of one of these systems.

```
ok probe-fcal-all
/pci@8,600000/SUNW,qlc@4
LiD HA  --- Port WWN ---      ---- Disk description ----
10  10  2100002037651b0e  SEAGATE ST318304FSUN18G 022D0017L007G2
12  12  2100002037651c12  SEAGATE ST318304FSUN18G 022D0017L007VJ
 1   1  2100002037653317  SEAGATE ST318304FSUN18G 032D0020L009TT
13  13  2100002037651f72  SEAGATE ST318304FSUN18G 022D0017L007JZ
11  11  2100002037651f76  SEAGATE ST318304FSUN18G 022D0017L007AL
14  14  2100002037651bf5  SEAGATE ST318304FSUN18G 022D0017L007XS
```

Identifying the System's Boot Device

Sun hardware uses the concept of a device tree to organize devices that are attached to the system.

Figure 8-6 shows the organizational structure of a device tree for an Ultra 5 or an Ultra 10 workstation.

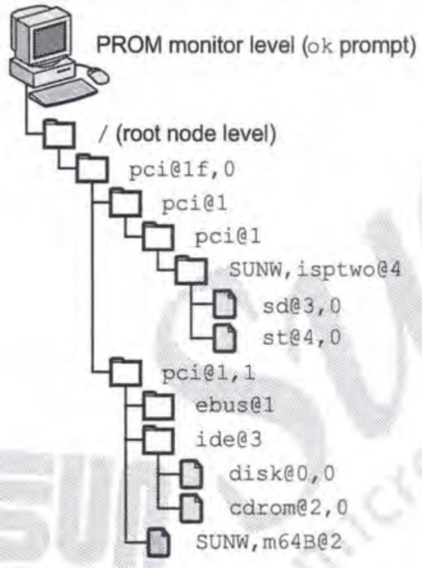

Figure 8-6 Partial Device Tree for an Ultra 5 or Ultra 10 Workstation

---

Note – In Figure 8-6, some license has been taken in naming these directories to simplify the illustration.

The OpenBoot firmware builds the device tree from information gathered at the POST. The device tree is loaded into memory and is used by the `kernel` during the boot process to identify all configured devices.

The top of the device tree is the root device node. Following the root device node is a bus nexus node. Connected to a bus nexus node is a leaf node, typically a controller for an attached device.

In Figure 8-6, the `disk@0,0` device is the IDE device for the hard disk drive, and the `cdrom@2,0` device is the IDE device for the CD-ROM drive. Both are attached to the IDE controller `ide@3`. Similarly, the `sd@3,0` device is the SCSI disk device and the `st@4,0` device is the SCSI tape device. Both are attached to the PCI-based SCSI controller `SUNW,isptwo@4`.

The paths built in the device tree by the OpenBoot firmware vary depending on the type of system and its device configuration.

Figure 8-7 shows a sample disk device path on an Ultra workstation with a PCI bus.

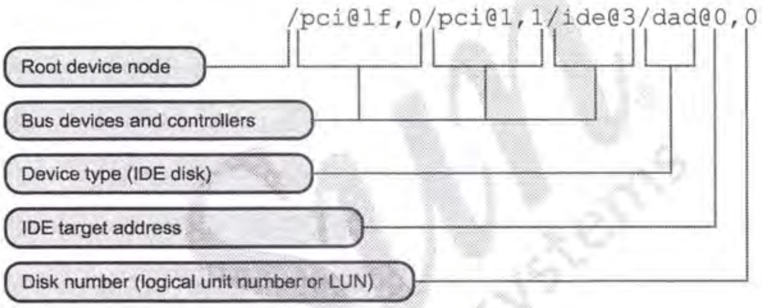

Figure 8-7 Disk Device Path – Ultra Workstation With a PCI IDE Bus

Figure 8-8 shows a sample disk device path on an Ultra workstation with a PCI-SCSI bus.

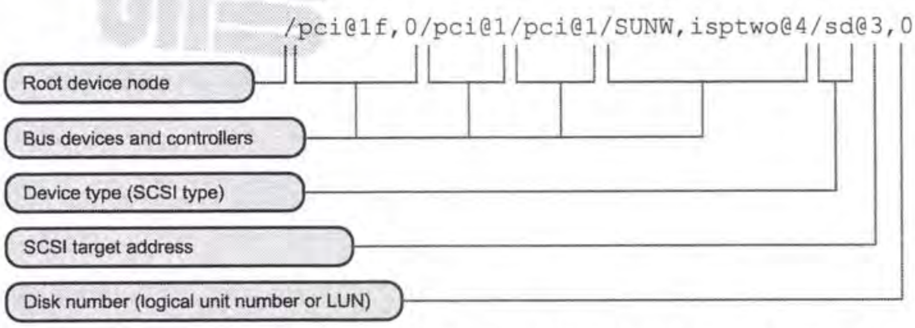

Figure 8-8 Disk Device Path – Ultra Workstation With a PCI-SCSI Bus

The show-devs Command

You use the show-devs command to view the entire device tree.

The following example shows output from the show-devs command.

```
ok show-devs
/SUNW,UltraSPARC-IIi@0,0
/pci@1f,0
/virtual-memory
/memory@0,10000000
/pci@1f,0/pci@1
/pci@1f,0/pci@1,1
/pci@1f,0/pci@1,1/ide@3
/pci@1f,0/pci@1,1/SUNW,m64B@2
/pci@1f,0/pci@1,1/network@1,1
/pci@1f,0/pci@1,1/ebus@1
/pci@1f,0/pci@1,1/ide@3/cdrom
/pci@1f,0/pci@1,1/ide@3/disk
/pci@1f,0/pci@1,1/ebus@1/SUNW,CS4231@14,200000
/pci@1f,0/pci@1,1/ebus@1/flashprom@10,0
/pci@1f,0/pci@1,1/ebus@1/eeprom@14,0
/pci@1f,0/pci@1/pci@1
/pci@1f,0/pci@1/pci@1/SUNW,isptwo@4
(output truncated)
ok
```

Note – In addition to the show-devs command, use the following additional OpenBoot PROM commands to view specific device information: show-ttys, show-displays, show-nets, show-disks, and show-tapes.

The `devalias` Command

To identify the current boot device alias for the system, use the `devalias` command.

The following example shows output from the `devalias` command.

```
ok devalias
screen          /pci@1f,0/pci@1,1/SUNW,m64B@2
net             /pci@1f,0/pci@1,1/network@1,1
cdrom           /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f
disk            /pci@1f,0/pci@1,1/ide@3/disk@0,0
disk3           /pci@1f,0/pci@1,1/ide@3/disk@3,0
disk2           /pci@1f,0/pci@1,1/ide@3/disk@2,0
disk1           /pci@1f,0/pci@1,1/ide@3/disk@1,0
disk0           /pci@1f,0/pci@1,1/ide@3/disk@0,0
ide             /pci@1f,0/pci@1,1/ide@3
floppy          /pci@1f,0/pci@1,1/ebus@1/fdthree
ttyb            /pci@1f,0/pci@1,1/ebus@1/se:b
ttya            /pci@1f,0/pci@1,1/ebus@1/se:a
keyboard!       /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8:forcemode
keyboard        /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse           /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
name            aliases
```

The left side of the command output lists the device alias names, and the right side of the output lists the physical address of each device.

Device aliases are hard-coded into the OpenBoot PROM firmware, and they are easier to remember and use than the physical device addresses. The `disk` device alias identifies the default boot device for the system.

The `boot-device` parameter sets the system's boot device in the NVRAM. By default, the `boot-device` parameter is set to `disk net`. You can view the system's boot device through commands from the `ok` prompt.

To boot the system from the default device, perform the `boot` command:

```
ok boot
```

Creating and Removing Custom Device Aliases

A portion of the NVRAM called NVRAMRC contains registers to hold custom parameters and is also reserved for storing new device alias names. External devices do not, by default, have built-in device aliases associated with them.

The NVRAMRC is affected by the commands `nvalias` and `nvunalias`, as well as the parameter `use-nvramrc?`.

The `nvalias` Command

You use the `nvalias` command to create a new device alias name to access a newly attached external device. The command format is:

```
nvalias aliasname device_path
```

The effect of the `nvalias` command is to store the following command line in the NVRAMRC:

```
devalias aliasname device_path
```

The following example shows how to add a new boot device alias, called `mydisk`, and boot the system from this new boot device alias.

Note – A shortcut provided with the `show-disks` command enables you to select a device and use the Control-Y keys to copy the device path onto the command line.

The example uses the `show-disks` command to select the device path for the disk being used. It then uses the `nvalias` command to create a new device alias called `mydisk`.

```
ok show-disks
a) /pci@1f,0/pci@1/scsi@1,1/disk
b) /pci@1f,0/pci@1/scsi@1/disk
c) /pci@1f,0/pci@1,1/ide@3/cdrom
d) /pci@1f,0/pci@1,1/ide@3/disk
e) /pci@1f,0/pci@1,1/ebus@1/fdthree@14,3023f0
q) NO SELECTION
Enter Selection, q to quit: d
/pci@1f,0/pci@1,1/ide@3/disk has been selected.
Type ^Y (Control-Y) to insert it in the command line.
```

```
e.g. ok nvalias mydev ^Y
        for creating devalias mydev for
/pci@1f,0/pci@1,1/ide@3/disk
ok nvalias mydisk ^y
```

To paste the device path for the selected disk, press Control-Y on the command line.

```
ok nvalias mydisk /pci@1f,0/pci@1,1/ide@3/disk@0,0
```

Note – When the device path has been pasted on the command line (by the Control-Y keys), the target number and logical unit number (LUN) must be added for the disk device, for example, `sd@0,0` or `disk@0,0`.

Set the `boot-device` parameter to the new value, in this case `mydisk`, and boot the system.

```
ok setenv boot-device mydisk
boot-device =         mydisk
ok boot
```

The `nvunalias` Command

You use the `nvunalias` command to remove an alias name.

To remove a custom device alias name, use the following command format:

```
ok nvunalias aliasname
```

Note – The `nvunalias` command is the single exception to the rule that changes to NVRAM occur immediately and do not require a `reset-all` command.

In the example, you would use the `nvunalias` command to delete the alias name `mydisk` from NVRAMRC and use the `setenv` command to set the `boot-device` parameter to `disk`.

```
ok nvunalias mydisk
ok setenv boot-device disk
boot-device =         disk
ok reset-all
Resetting ...
```

Viewing and Changing NVRAM Parameters From the Shell

Use the /usr/sbin/eeprom command to view and to change the NVRAM parameters while the Solaris OE is running.

Using the eeprom Command

Be aware of the following guidelines when using the eeprom command:

- Only the root user can change the value of a parameter.
- You must enclose parameters with a trailing question mark in single quotation marks (single quotes) when the command is executed in the C shell.
- All changes are permanent. You cannot run a reset command to undo the parameter changes.

The following examples use the eeprom command to view and change NVRAM parameters.

- To list all of the parameters with their current values, perform the command:

```
# eeprom
```

- To list a single parameter and its value, in this case, the boot-device parameter, perform the command:

```
# eeprom boot-device
boot-device=disk
#
```

- To change the value of the default boot device to disk2, perform the command:

```
# eeprom boot-device=disk2
#
```

- To change the value of the auto-boot? parameter, perform the command:

```
# eeprom auto-boot?=true
#
```

Interrupting an Unresponsive System

When a system freezes or stops responding to the keyboard, you might have to interrupt it. When you interrupt the system, all active processes stop immediately, and the processor services the OpenBoot PROM exclusively. It does not allow you to flush memory or to synchronize file systems.

Aborting an Unresponsive System

To abort or interrupt an unresponsive system:

1. Attempt a remote login on the unresponsive system to locate and kill the offending process.
2. Attempt to reboot the unresponsive system gracefully.
3. Hold down the Stop-A key sequence on the keyboard of the unresponsive system. The system is placed at the `ok` prompt.

Note – If an ASCII terminal is being used as the system console, use the Break sequence keys.

---

4. Manually synchronize the file systems by using the OpenBoot PROM `sync` command.

```
ok sync
```

This command causes the syncing of file systems, a crash dump of memory, and then a reboot of the system.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine the commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using the OpenBoot PROM Commands (Level 1)

In this exercise, you use the OpenBoot PROM and Solaris OE commands to perform the tasks described in this module.

Preparation

Refer to the lecture notes as necessary to perform the following tasks and answer the questions listed.

Tasks

Complete the following tasks:

- Shut down the system to run level 0, and gather information about your system. Find out the following:
  - OpenBoot PROM revision
  - Megabytes of installed memory
  - System type
  - NVRAM serial number
  - Ethernet address
  - Host ID
- Set the `auto-boot?` parameter to `false`.

  (Steps 1–11 in the Level 2 lab)
- Create a new device alias called `mydisk` that uses the same device as the `disk` device alias. Verify the contents of the `nvramrc` file, and verify how to set the `use-nvramrc?` parameter.

  (Steps 12–17 in the Level 2 lab)
- Boot the system using the new alias. As the `root` user, use the `eeprom` command to list all parameters. Set the `boot-device` parameter to the `mydisk` device alias.

  (Steps 18–22 in the Level 2 lab)

- Shut down the system to run level 0, and verify the change you made by using the `printenv` command. Remove the `mydisk` device alias. Reset the `boot-device` parameter to its default value, and boot the system.

  (Steps 23–31 in the Level 2 lab)

Exercise: Using the OpenBoot PROM Commands (Level 2)

In this exercise, you use the OpenBoot PROM and Solaris OE commands to perform the tasks described in this module.

Preparation

Refer to the lecture notes as necessary to perform the following tasks and answer the questions listed.

Task Summary

In this exercise, you accomplish the following:

- Shut down the system to run level 0, and use the following commands to set parameters and gather basic information about your system.

  ```
  banner
  set-defaults
  help
  help file
  printenv
  setenv
  reset-all
  probe-scsi
  probe-scsi-all
  probe-ide
  ```
- Set the `auto-boot?` parameter to `false`.
- Create a new device alias called `mydisk` that uses the same device as the `disk` device alias. Verify the contents of the `nvramrc` file, and verify how to set the `use-nvramrc?` parameter.
- Boot the system using the new alias. As the `root` user, use the `eeprom` command to list all parameters. Set the `boot-device` parameter to the `mydisk` device alias.
- Shut down the system to run level 0, and verify the change you made by using the `printenv` command. Remove the `mydisk` device alias. Reset the `boot-device` parameter to its default value, and boot the system.

Tasks

Complete the following steps:

1. If the Solaris OE is currently running, log in as the `root` user, and halt your system.
2. When the `ok` prompt appears, set the OpenBoot PROM parameters to their default values.
3. Use the `help` command to display the list of help topics.
4. Use the `help` command to display information about the `boot` command.

   What does the `help` command list for `boot`?
5. Use the `banner` command to obtain the following information:

   OpenBoot PROM revision:

   Megabytes of installed memory:

   System type:

   NVRAM serial number:

   Ethernet address:

   Host ID:
6. Use the `printenv` command to display the list of OpenBoot PROM parameters. Record the current values for the following parameters:

   `output-device`
   `input-device`
   `auto-boot?`
   `boot-device`
7. Prevent the system from booting automatically after you use the `reset-all` command by setting the `auto-boot?` parameter to `false`.
8. Use the `reset-all` command to verify that the new `auto-boot?` value is in effect. The system should remain at the `ok` prompt after the `reset-all` command completes.
9. Use the `probe-scsi`, `probe-scsi-all`, and `probe-ide` commands to display the list of disk devices attached to your system. Not all of these commands are present on all systems.
10. What are the main differences that you see in the information that these commands display?

11. List the target number and device type (disk, tape, or CD-ROM) of all the devices shown by the `probe-scsi`, `probe-scsi-all`, and `probe-ide` commands.
12. Verify that your default `boot-device` is set to `disk net`.
13. Use the `devalias` command to display the full device path for the `disk` alias.

    Record the path name reported:
14. Use the `show-disks` command to select the device path that relates to the disk recorded in Step 13, and use the `nvalias` command to create a new device alias called `mydisk`. Set the `mydisk` alias to the path and disk name you recorded in Step 13.

    Remember to use the Control-Y key sequence to paste the disk path into your `nvalias` command. You must manually complete the path to specify the disk you want to use.
15. Verify that the new alias is correctly set.
16. Use the `printenv` command to display the contents of the `nvramrc` file.

    What command does the `nvramrc` file contain that creates the `mydisk` alias?
17. Use the `printenv` command to display the setting of the `use-nvramrc?` parameter.

    What is the current setting of the `use-nvramrc?` parameter?
18. Boot your system using the `mydisk` alias.
19. Log in as the `root` user on your system. Open a new terminal window.
20. Use the `eeprom` command to list all NVRAM parameters.
21. Use the `eeprom` command to list the setting of the `boot-device` parameter.
22. Use the `eeprom` command to set the `boot-device` parameter to the alias `mydisk`.
23. Bring your system to run level 0.
24. Verify that the `eeprom` command set the `boot-device` parameter to the alias `mydisk`.
25. Set the `boot-device` parameter to its default value, and verify the setting.
26. Use the `nvunalias` command to remove the alias `mydisk`.

27. Verify that the mydisk alias is no longer in the nvramrc file.
28. Use the devalias command to see if the mydisk alias has been removed from the list of device aliases.

    Has it?
29. Run the reset-all command, and then check again if the mydisk alias has been removed from the list of device aliases.

    (If your system reboots, interrupt the reboot with a Stop-A key sequence.)

    Has it?
30. Set the OpenBoot PROM parameters back to their default values, and boot the system from the default device.
31. Log in as the root user.

Exercise: Using the OpenBoot PROM Commands (Level 3)

In this exercise, you use the OpenBoot PROM and Solaris OE commands to perform the tasks described in this module.

Preparation

Refer to the lecture notes as necessary to perform the following tasks and answer the questions listed.

Task Summary

In this exercise, you accomplish the following:

- Shut down the system to run level 0, and use the following commands to set parameters and gather basic information about your system.

  ```
  banner
  set-defaults
  help
  help file
  printenv
  setenv
  reset-all
  probe-scsi
  probe-scsi-all
  probe-ide
  ```
- Set the `auto-boot?` parameter to `false`.
- Create a new device alias called `mydisk` that uses the same device as the `disk` device alias. Verify the contents of the `nvramrc` file, and verify how to set the `use-nvramrc?` parameter.
- Boot the system using the new alias. As the `root` user, use the `eeprom` command to list all parameters. Set the `boot-device` parameter to the `mydisk` device alias.
- Shut down the system to run level 0, and verify the change you made by using the `printenv` command. Remove the `mydisk` device alias. Reset the `boot-device` parameter to its default value, and boot the system.

Tasks and Solutions

Complete the following steps:

1. If the Solaris OE is currently running, log in as the `root` user, and halt your system.

```
# init 0
```

2. When the `ok` prompt appears, set the OpenBoot PROM parameters to their default values.

```
ok set-defaults
```

3. Use the `help` command to display the list of help topics.

```
ok help
```

4. Use the `help` command to display information about the `boot` command.

```
ok help boot
```

What does the `help` command list for `boot`?

`boot` – *Default boot (values specified in NVRAM variables).*

`boot <network-device>:[dhcp,][server-ip], [boot-filename], [client-ip], [router-ip], [boot-retries], [tftp-retries], [subnet-mask], [boot-arguments].`

5. Use the `banner` command to obtain the following information:

OpenBoot PROM revision:

Megabytes of installed memory:

System type:

NVRAM serial number:

Ethernet address:

Host ID:

Each system presents its own unique information.

6. Use the `printenv` command to display the list of OpenBoot PROM parameters. Record the current values for the following parameters:

```
ok printenv
output-device  - screen
input-device    - keyboard
auto-boot?      - true
boot-device     - disk net
```

7. Prevent the system from booting automatically after using the `reset-all` command by setting the `auto-boot?` parameter to `false`.

```
ok setenv auto-boot? false
```

8. Use the `reset-all` command to verify that the new `auto-boot?` value is in effect. The system should remain at the `ok` prompt after the `reset-all` command completes.

```
ok reset-all
```

9. Use the `probe-scsi`, `probe-scsi-all`, and `probe-ide` commands to display the list of disk devices attached to your system. Not all of these commands are present on all systems.

```
ok probe-scsi
ok probe-scsi-all
ok probe-ide
```

10. What are the main differences that you see in the information that these commands display?

    The `probe-scsi-all` *command lists all devices on all SCSI chains and their full device paths. The* `probe-scsi` *command only lists devices on the first SCSI chain and does not list the full device paths. The* `probe-ide` *command reports the list of IDE devices attached to the system.*

11. List the target number and device type (disk, tape, or CD-ROM) of all the devices shown by the `probe-scsi`, `probe-scsi-all`, and `probe-ide` commands.

    Each system presents its own unique information.

12. Verify that your default `boot-device` is set to `disk net`.

```
ok printenv boot-device
```

13. Use the `devalias` command to display the full device path for the `disk` alias.

```
ok devalias disk
```

    Record the path name reported:

    This differs from system to system. On an Ultra 5 workstation, the alias is defined as follows:

    `/pci@1f,0/pci@1,1/ide@3/disk@0,0`

14. Use the `show-disks` command to select the device path that relates to the disk recorded in Step 13, and use the `nvalias` command to create a new device alias called `mydisk`. Set the `mydisk` alias to the path and disk name you recorded in Step 13.

Remember to use the Control-Y key sequence to paste the disk path into your `nvalias` command. You must manually complete the path to specify the disk you want to use.

```
ok show-disks
(select one of the disks from the list)
ok nvalias mydisk pathname@#,#
```

15. Verify that the new alias is correctly set.

```
ok devalias mydisk
```

16. Use the `printenv` command to display the contents of the `nvramrc` file.

```
ok printenv nvramrc
```

What command does the `nvramrc` file contain that creates the `mydisk` alias?

Systems differ according to the disk devices they use. An Ultra 5 workstation would report the following:

```
devalias mydisk /pci@1f,0/pci@1,1/ide@3/disk@0,0
```

17. Use the `printenv` command to display the setting of the `use-nvramrc?` parameter.

```
ok printenv use-nvramrc?
```

What is the current setting of the `use-nvramrc?` parameter?

`true`

18. Boot your system using the `mydisk` alias.

```
ok boot mydisk
```

19. Log in as the `root` user on your system. Open a new terminal window.

20. Use the `eeprom` command to list all NVRAM parameters.

```
# eeprom
```

21. Use the `eeprom` command to list the setting of the `boot-device` parameter.

```
# eeprom boot-device
```

22. Use the `eeprom` command to set the `boot-device` parameter to the alias `mydisk`.

```
# eeprom boot-device=mydisk
```

23. Bring your system to run level 0.

```
# init 0
```

24. Verify that the `eeprom` command set the `boot-device` parameter to the alias `mydisk`.

```
ok printenv boot-device
```

25. Set the `boot-device` parameter to its default value, and verify the setting.

```
ok set-default boot-device
ok printenv boot-device
```

26. Use the `nvunalias` command to remove the alias `mydisk`.

```
ok nvunalias mydisk
```

27. Verify that the `mydisk` alias is no longer in the `nvramrc` file.

```
ok printenv nvramrc
```

28. Use the `devalias` command to see if the `mydisk` alias has been removed from the list of device aliases.

```
ok devalias mydisk
```

Has it?

No

29. Run the `reset-all` command, and then check again if the `mydisk` alias has been removed from the list of device aliases.

```
ok reset-all
```

(If your system reboots, interrupt the reboot with a Stop-A key sequence.)

```
ok devalias mydisk
```

Has it?

Yes

30. Set the OpenBoot PROM parameters back to their default values, and boot the system from the default device.

```
ok set-defaults
ok reset-all
```

31. Log in as the `root` user.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

SUN MICROSYSTEMS | EDUCATIONAL SERVICES

STUDENT GUIDE

SUN EDUCATIONAL SERVICES

Intermediate System
Administration for the Solaris™ 9
Operating Environment
SA-239

Revision A.2

VOL I

Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303-4900 USA 1-650-960-1300 or 1-800-555-9sun www.sun.com/training